

Data-Driven Optimization and Statistical Modeling to Improve Meter Reading for Utility Companies

Citation for published version (APA):

Sinha Roy, D., Defryn, C., Golden, B., & Wasil, E. (2019). *Data-Driven Optimization and Statistical Modeling to Improve Meter Reading for Utility Companies*. Maastricht University, Graduate School of Business and Economics. GSBE Research Memoranda No. 028
<https://doi.org/10.26481/umagsb.2019028>

Document status and date:

Published: 12/12/2019

DOI:

[10.26481/umagsb.2019028](https://doi.org/10.26481/umagsb.2019028)

Document Version:

Publisher's PDF, also known as Version of record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

Download date: 04 May. 2023

Debdatta Sinha Roy,
Christof Defryn, Bruce golden,
Edward Wasil

**Data-Driven Optimization and
Statistical Modeling to
Improve Meter Reading for
Utility Companies**

RM/19/028

GSBE

Maastricht University School of Business and Economics
Graduate School of Business and Economics

P.O Box 616
NL- 6200 MD Maastricht
The Netherlands

Data-Driven Optimization and Statistical Modeling to Improve Meter Reading for Utility Companies

Debdatta Sinha Roy¹, Christof Defryn², Bruce Golden¹, and Edward Wasil³

¹Robert H. Smith School of Business, University of Maryland, College Park, MD 20742

²School of Business and Economics, Maastricht University, 6211 LM Maastricht,
Netherlands

³Kogod School of Business, American University, Washington, DC 20016

Abstract

Utility companies collect usage data from meters on a regular basis. The usage data are collected automatically using radio-frequency identification (RFID) technology. Each meter transmits signals from an RFID tag that are read by a vehicle-mounted reading device within a specified distance. Routing the vehicles can be modeled by a close enough vehicle routing problem on a street network. In practice, there is uncertainty while reading meters. The signal transmitted by an RFID tag is discontinuous, and the range that each meter can be read is different and stochastic due to weather conditions, surrounding obstacles, interference, and decreasing battery life of the RFID tags. These factors can lead to meters not being read. A vehicle has to be sent at a later time to read the missed meters, and this leads to increased costs for a utility company due to additional operational costs and overtime payments to drivers. Our aim is to address the uncertainty issues of the RFID technology by generating routes that are both cost-effective and robust (we seek to minimize the number of missed reads). We use data analytics, optimization, and Bayesian statistical models to address the uncertainty. Simulation experiments using real data show that the hierarchical Bayesian statistical model gives better results compared to other Bayesian statistical models. Utility companies can potentially integrate the results from the hierarchical Bayesian statistical model into their route generating software as a decision-support tool to produce routes that are more cost-effective and robust than the routes that they currently generate.

Keywords: Bayesian statistics, Integer programming, Vehicle routing, Meter reading

1 Introduction

1.1 Background and Literature Review

Utility companies have to read many thousands of electric, gas, and water meters of residential and commercial customers on a regular monthly basis. Meters are read automatically using a short distance wireless technology called radio-frequency identification (RFID) technology.

An automatic meter reading (AMR) system has two parts: an RFID tag and a vehicle-mounted reading device. An RFID tag is connected to a physical meter. The tag encodes the identification number of the meter and its current reading into a digital signal. The vehicle-mounted reading device collects the data automatically when it approaches an RFID tag within a specified distance. Utility companies would like to design the routes of the vehicles to cover all customers (meters) in the service area while minimizing the total cost or length of the routes. The use of RFID technology in meter reading changes the routing problem from a standard

Table 1: A brief chronological summary of the meter reading problem.

Early 1960s	RFID technology was first tested by AT&T in cooperation with a group of utility companies and Westinghouse.
Late 1970s	Stern and Dror (1979) made one of the earliest efforts to solve the meter reading problem. They developed a route-first, cluster-second approach.
Early 1990s	Geographic data became available in the form of GBF/DIME (geographic base file/dual independent map encoding) files. This led to the development of optimization algorithms, graphics, and interactive features in meter-reader software systems. The computerized routing system of Bodin and Levy (1991) produced much better routes than the routes generated by utility companies.
Early 2000s	Geographic information systems (GISs) were combined with powerful (near-optimal) routing algorithms to form highly visual computerized systems. Levy et al. (2002) described how several layers of data can be displayed in a service area by a GIS. The displayed layers can then be used to select a subset of meters to read within the service area for route planning purposes.
Late 2000s	RFID technology was used extensively in many industries for tracking resources since it holds down cost while increasing accuracy compared to traditional labor-intensive reading methods. The accuracy of transmitters and receivers improved and the cost decreased gradually with the advancement of technology, making RFID technology even more viable and useful. Eglese et al. (2014) summarized the meter reading problem from the late 1970s until the late 2000s.

vehicle routing problem (VRP) to a close-enough VRP (CEVRP). Substantial savings over traditional solutions are possible by developing routes that exploit this close-enough feature, i.e., the meter readers only have to be within a specified distance from the meters to read them and not manually visit each one. Table 1 gives a brief chronological summary of the meter reading problem.

Most of the research on the close-enough problem is limited to Euclidean distances and uses a node routing formulation. Dumitrescu and Mitchell (2003) studied approximation algorithms for the close-enough traveling salesman problem (CETSP). Gulczynski et al. (2006) and Dong et al. (2007) presented clustering and convex hull heuristics for the CETSP in the context of meter reading. Mennell (2009) and Behdani and Smith (2014) formulated mixed integer programs for the CETSP. Coutinho et al. (2016) proposed an exact algorithm for the CETSP based on a branch-and-bound procedure and second order cone programming. Groër et al. (2009) addressed the balanced billing cycle vehicle routing problem (BBCVRP) which occurs when, over time, routes become inefficient and fractured with imbalanced workloads for the meter readers. Their three-stage algorithm for solving the BBCVRP used partitioning heuristics and integer programming to reduce the length of the routes and to balance the workload.

Shuttleworth et al. (2008) were the first to model the CETSP with an arc routing formulation. They developed a two-stage process to solve the CETSP over a street network for a single meter-reader route. In the first stage, two heuristics (weighted bang for buck, distance weighted bang for buck) and two integer programs specify a subset of street segments that have to be traversed by a meter reader. All meters are within distance r from at least one location on at least one of the specified street segments. In the second stage, a travel path (cycle) is generated that traverses the specified street segments. Hà et al. (2014) proposed mathematical formulations and heuristics for the close-enough arc routing problem (CEARP). In the CEARP, traversed street segments only have to be within a specified distance from the points of interest. Ávila et al. (2016) proposed a new mathematical formulation for the CEARP and described its polyhedra. Renaud et al. (2017) considered a version of the CEARP in the context of meter reading in which the probability of reading a meter from a street segment decays exponentially as the distance from the meter to the street segment increases. They proposed an integer programming formulation and presented several heuristics.

There are issues with RFID technology that are not considered in the literature that we need

to take into account. The signal transmitted by an RFID tag occurs at regular time intervals that are not continuous. This is done to extend the battery life of the RFID tags. This leads to the possibility of a missed capture of a signal if the vehicle with the receiver is within the range of the meter only for a short time. Also, the signal range of a meter can vary from the distance specified by the manufacturers of the RFID devices due to weather conditions, surrounding obstacles, signal interference from other meters in the vicinity, and decreasing battery life of the RFID tags.

On average, utility companies read more than 1.5 million RFID meters on a monthly basis. It is observed that around 5-10% of those meters are missed from the planned routes of the meter reading vehicles. Currently, utility companies generate meter reading routes by solving the CEVRP such that all meters are within a specified distance (range of the signal as specified by the manufacturers of the RFID devices) from at least one location on the route. Utility companies always make a special attempt to read the missed meters for commercial and industrial customers because these tend to generate higher revenues. For residential customers, they want to use estimated billing for the missed meters. However, the public utility commission in many areas will not allow estimated billing. For example, in Illinois, utility companies have to perform actual meter reading at least every second billing cycle (Illinois Administrative Code 2018). Similar examples can be found in Colorado (Colorado Department of Regulatory Agencies 2018), Michigan (Michigan Department of Labor and Economic Growth 2018), and Irving, Texas (Irving, Texas - Code of Ordinances 2018). A vehicle has to be sent at a later time to read the missed meters, and this leads to increased costs due to additional operational costs and overtime payments to drivers.

1.2 Research Goal and Contributions of the Paper

In the meter reading context, we will address the above-mentioned issues of the RFID technology by generating routes for the CEVRP that are both cost-effective and robust (in the sense that we seek to minimize the number of missed reads). This is done by bringing together data analytics, statistical modeling, and optimization techniques. The idea is to significantly reduce the number of missed meters even though the routes that are generated may be somewhat longer than those currently used by a utility company. For the utility companies, it is much easier and cost-effective if they know ex-ante that they have to traverse a somewhat longer route that

leads to fewer missed meters. This substantially reduces the need to dispatch a vehicle to read the missed meters that may be spread throughout the street network. While past research has focused on mathematical formulations and computational experiments on artificially generated networks, we use real networks and actual meter reading data from utility companies to solve a more realistic version of this problem. Real networks that we use are a few orders of magnitude larger than the artificial networks used in the literature. The most important factor is that the way in which street segments and meters are distributed on real networks, is very different from that in artificial networks. Hà (2012) gives a detailed description of how artificial meter reading networks are systematically generated. Therefore, the computational performances of the heuristics discussed in the literature do not have enough practical relevance.

We summarize the main contributions of our paper as follows.

1. We formulate the stochastic meter reading problem as a two-stage integer program (IP), where the Stage 1 IP is a linear IP that guarantees a pre-specified likelihood of reading the meters. The Stage 2 IP adds deadhead segments to the solution of the Stage 1 IP to generate the full route. The two-stage IP formulation is deterministic even though the use of the RFID technology makes the meter reading problem inherently stochastic.
2. We develop three Bayesian updating learning models, namely, a logit model, a probit model, and a hierarchical probit model to capture the uncertainty in the data and also to avoid the shortcomings of regression. We show that the hierarchical probit model gives a more accurate estimate of the probability that a meter is read successfully compared to logit and probit models. We perform simulation experiments using an actual street network with meter locations to show that the hierarchical probit model generates robust routes, i.e., the number of missed meters is significantly less compared to the other two Bayesian models, even though the routes may be slightly longer.
3. We present an iterative algorithmic framework. We start by learning from the incoming data every time the meter reading vehicle collects readings. We then re-solve the two-stage IP with the updated probability of a meter being read successfully to generate routes that are more robust for addressing the uncertainty. Utility companies can integrate this algorithm into their route generating software as a decision-support tool.

Table 2: Summary of the data set.

Total number of meters in the service location layer	6,067
Number of meters in the service location layer that are read	5,720
Total number of read events	337,870
Total number of street segments in the network	1,575
Number of street segments traversed in the route	578
Number of street segments traversed in the route counting repetitions	829
Total number of nodes in the network	1,072
Duration of the route (hours)	6
Time gap between consecutive signal transmission (sec)	3

1.3 Structure of the Paper

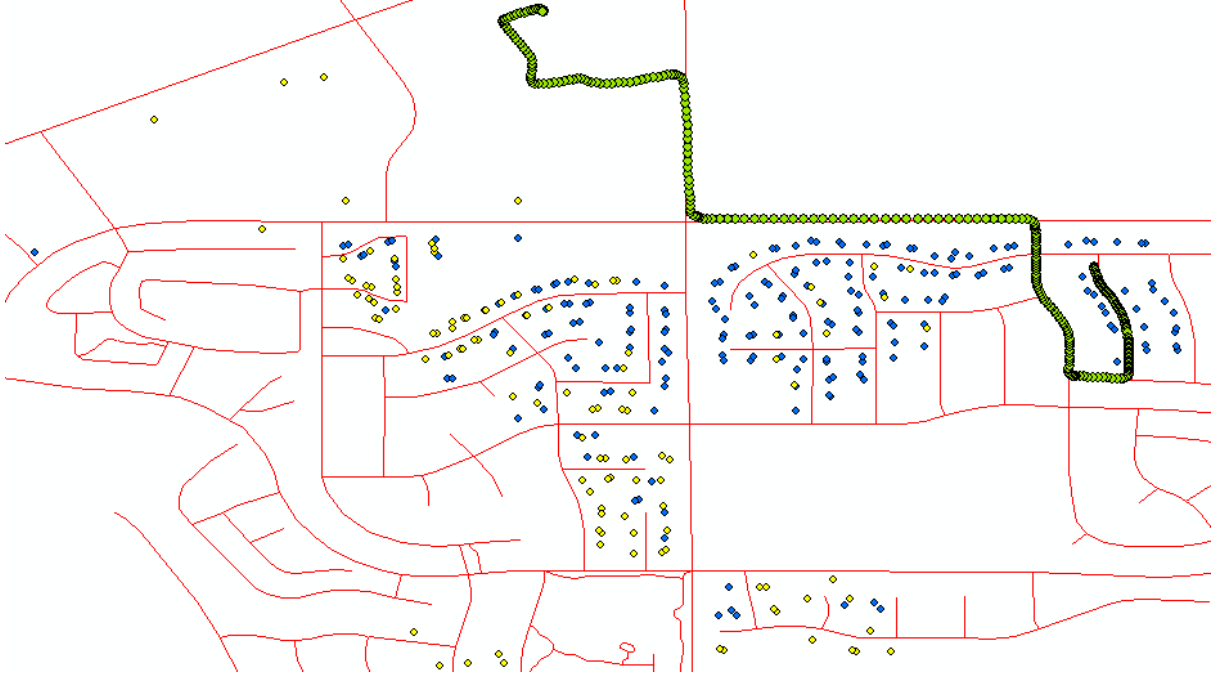
The remainder of the paper is organized as follows. In Section 2, we describe the size and content of the data set. In Section 3, we give the mathematical programming formulation for our problem. In Section 4, we conduct computational experiments to show the performance of the Stage 1 IP. In Section 5, we describe the metaheuristic for solving the Stage 2 IP. In Section 6, we develop the Bayesian updating learning models to capture the uncertainty in the data. In Section 7, we conduct simulation experiments using our data set to compare and quantify the performance of the Bayesian updating learning models and also describe our decision-support tool. In Section 8, we summarize our contributions and provide future research directions.

2 Description of the Data Set

The data set was gathered during the first half of 2016 by ITRON (a technology and services company that manufactures RFID equipment) and provided by RouteSmart Technologies. Table 2 summarizes the data set which gives meter locations and reading data serviced by Connecticut Natural Gas in Hartford, Connecticut. The data are in GIS format and have three layers.

1. Street Level Data. Information about the shape, length, and type of street segments.
2. Service Location Data. Geographic locations of all meters that should be read and the unique account identifier of each meter.
3. Reading Events Data. Records of all read events by the meter reading vehicle in the form of the time of read (with a resolution of one second), the account identifier of the meter that is read, and the geographic location of the vehicle during the read.

Figure 1: (Color online) A view of the street layer, the service location layer, and the reading events layer. The red lines represent the street segments. The green dots represent a portion of the route traversed by the meter reading vehicle. The blue dots and the yellow dots represent meters (customers) in the service location layer that are read and that are missed, respectively. If multiple meters have the same geographic location, then they are represented by a single dot, although they have distinct account identifiers.



The data are represented using ArcGIS (Steiniger and Hunter 2013). In Figure 1, we show how the data appear in GIS format with views of the street layer, the service location layer, and the reading events layer. After the vehicle has traversed a portion of the route marked by the green dots, Figure 1 shows the meters in the service location layer that are read (blue dots) and those that are missed (yellow dots). Even though ITRON specifies the range of the RFID signals to be around 500 feet, some missed meters are well within that range, while some meters that are read are well outside of it. The routes generated should address these variabilities.

From the data, we make the following observations. There are many account identifiers in the reading events file that have no corresponding entry in the service location file, i.e., the RFID readers are picking up signals from nearby RFID tags that do not require reading by the utility company. There are a total of 337,870 read events from a route that took six hours and traversed 829 street segments (counting repetitions). Many of those read events are from unwanted RFID tags. The read events data have a many-to-one relationship to a service

location account identifier, i.e., some of the meters are read more than once by the meter reading device. Out of the 6,067 meters in our data set, 347 meters are missed. The vehicle location is tracked every second. However, when the read events for a single meter are recorded, they do not occur every second along a street segment that seems to be within range. Rather, there is generally a regular time gap between occurrences of read events for the same meter. This confirms the fact that the signal transmitted by an RFID tag is at regular time intervals and is not occurring continuously. Some meters that are very close to the vehicle route have been missed, probably due to a discontinuous signal. There is a time gap of three seconds between successive signal transmissions from the RFID tags in our data set. Missed reads can also be due to the variability of the range of a meter to transmit a signal.

3 Integer Programming Formulation

We formulate the meter reading problem with RFID technology as a two-stage IP. The Stage 1 IP finds the street segments that are to be traversed for reading each meter with a pre-specified chance of being read. The solution of the Stage 1 IP gives street segments spread across the street network, which does not necessarily form a full route. A mixed rural postman problem finds the shortest way of connecting a given set of required street segments to form a full route on a mixed graph with edges (two-way street segments) and arcs (one-way street segments). The Stage 2 IP solves a mixed rural postman problem that adds deadhead segments (extra street segments not required for reading meters) to the solution of the Stage 1 IP to obtain the full route and it ensures that the depot (denoted by a node on the graph) is a part of the route.

3.1 Stage 1 IP Formulation

Consider a street network as a mixed graph denoted by $G = (V, E \cup A)$, where E denotes the set of the edges, A denotes the set of the arcs, and V denotes the set of nodes. Let $c_j \geq 0$ be the cost (length) of street segment j . Let I be the set of the meters. Let p_{ij} be the probability that meter i is read at least once from street segment j . Let $L_i \in [0, 1]$ be the specified likelihood of reading meter i from the full route. We define x_j to be the binary decision variable denoting whether or not street segment j should be traversed. The Stage 1 IP formulation is given by

the following.

$$\text{(Stage 1 IP)} \quad \min \quad \sum_{j \in E \cup A} c_j x_j \quad (1)$$

$$\text{s.t.} \quad \prod_{j \in E \cup A} (1 - p_{ij})^{x_j} \leq (1 - L_i) \quad \forall i \in I \quad (2)$$

$$x_j \in \{0, 1\} \quad \forall j \in E \cup A \quad (3)$$

The objective function (1) minimizes the total cost (length). Constraints (2) select the values of the binary decision variables (x_j) so that the probability of reading meter i is at least L_i . Constraints (3) define the decision variables. In general, the solution of the Stage 1 IP, i.e., the graph induced by the required edges and arcs $G_R = (V, E_R \cup A_R)$, where $E_R \subseteq E$ and $A_R \subseteq A$ denote the set of required edges and arcs, respectively, is not connected. The objective value of the Stage 1 IP will be greater for larger values of L_i . The greater the need to read meter i during the next meter reading trip, the larger should be the value of L_i set by the utility company. In cases where the utility company can manage using estimated billing for meter i during the next billing cycle, the value of L_i should be set close to 0. We note that constraints (2) can be linearized in the decision variables $\sum_{j \in E \cup A} x_j \times \log(1 - p_{ij}) \leq \log(1 - L_i)$ for all meters i yielding a linear Stage 1 IP.

For values of L_i close to 1, constraints (2) can be infeasible for some meter i even when the meter reading vehicle traverses all street segments in the network ($x_j = 1$ for all street segments j), i.e., meter i cannot be read automatically with probability of at least L_i . In that case, the driver of the meter reading vehicle will need to park the vehicle on the closest street segment and read meter i manually. This means that meter i is read with probability 1 from the closest street segment, i.e., the Stage 1 IP is solved with $p_{ij} = 1$, where j is the closest street segment to meter i . This will enforce $x_j = 1$ in the Stage 1 IP solution, and, therefore, street segment j will be in the set of required street segments. Let $M_R \subseteq E_R \cup A_R$ denote the subset of the required street segments that are needed to manually read some of the meters, i.e., $p_{ij} = 1$ for all $j \in M_R$. We consider a constant stoppage time to manually read meter i from street segment j . Accordingly, we add a penalty to the Stage 2 IP objective value as a proxy for the distance that could have been traversed during the stoppage time.

3.2 Stage 2 IP Formulation

For $S_1, S_2 \subseteq V$, $(S_1 : S_2)$ denotes the set of edges and arcs with one endpoint in S_1 and the other endpoint in S_2 . $A(S_1 : S_2) = \{(i, j) \in A : i \in S_1, j \in S_2\}$ denotes the set of arcs with one endpoint in S_1 and the other endpoint in S_2 . $E(S_1 : S_2) = \{(i, j) \in E : i \in S_1, j \in S_2\}$ denotes the set of edges with one endpoint in S_1 and the other endpoint in S_2 . For $S \subseteq V$, $\delta^+(S) = A(S : V \setminus S)$, $\delta^-(S) = A(V \setminus S : S)$ and $\delta(S) = E(S : V \setminus S)$, where $E(S)$ and $A(S)$ denote the set of edges and arcs, respectively, with both endpoints in S . $\delta^*(S) = \delta(S) \cup \delta^+(S) \cup \delta^-(S) = (S : V \setminus S)$. If $S = \{v_i\}$, we simply write $\delta(i)$, $\delta^+(i)$, $\delta^-(i)$ or $\delta^*(i)$. The vertex sets of the connected components of G_R are denoted by V_1, \dots, V_p . The depot is denoted by the node $v_0 \in V$. We consider a single meter reading vehicle. We define y_j to be the non-negative integer decision variable denoting the number of times street segment j is traversed in the full route. For $F \subseteq E \cup A$, $Y(F) = \sum_{j \in F} y_j$. The Stage 2 IP formulation is given by the following.

$$\text{(Stage 2 IP)} \quad \min \quad \sum_{j \in E \cup A} c_j y_j \quad (4)$$

$$\text{s.t.} \quad Y(\delta^*(0)) \geq 1 \quad (5)$$

$$Y(\delta^*(i)) \equiv 0 \pmod{2} \quad \forall i \in V \quad (6)$$

$$Y(\delta^+(S)) \geq 1 \quad \forall S = \cup_{k \in Q} V_k, \quad Q \subset \{1, \dots, p\} \quad (7)$$

$$Y(\delta^+(S)) - Y(\delta^-(S)) \leq Y(\delta(S)) \quad \forall S \subset V \quad (8)$$

$$y_j \geq 1 \text{ and integer} \quad \forall j \in E_R \cup A_R \quad (9)$$

$$y_j \geq 0 \text{ and integer} \quad \forall j \in E \cup A \setminus E_R \cup A_R \quad (10)$$

The objective function (4) minimizes the total cost (length) of the route. Constraint (5) ensures that the depot is a part of the route. Constraints (6) are the flow conservation constraints, i.e., every node has an even degree in the route. Constraints (7) are the disjoint subtour elimination constraints, i.e., the required street segments obtained in the Stage 1 IP are connected in the route. Constraints (8) are the balanced-set inequalities, i.e., the difference between the number of arcs in the route entering S and the number of arcs in the route leaving S cannot be more than the number of edges in the route between S and $V \setminus S$. Constraints (9) define the decision

variables for those street segments j which are required to be traversed by the Stage 1 IP, i.e., $x_j = 1$. Constraints (10) define the decision variables for those street segments j which are not required to be traversed by the Stage 1 IP, i.e., $x_j = 0$. The Stage 1 IP solution already meets the specified likelihood L_i of reading each meter i . The deadhead segments added in the Stage 2 IP increase the likelihood of reading the meters because the meter reading vehicle is also receiving signals while traversing the deadhead segments. The Stage 2 IP formulation without constraint (5) is the formulation for the mixed rural postman problem (Corberán et al. 2014).

4 Computational Experiments for the Linear Stage 1 IP

We perform computational experiments that are designed to examine the performance of the linear Stage 1 IP and how performance varies with respect to the size of the data set. Understanding the performance of the linear Stage 1 IP on a large data set is important because our data set has 6,067 meters and 1,575 street segments.

We test the linear Stage 1 IP for different values of the number of meters ($|I|$) and the number of street segments ($|E \cup A|$), where the range of both values is 10 to 2000. The probabilities (p_{ij}) are generated randomly from a Uniform (0, 1) distribution. The costs (c_j) of the street segments are generated from three different distributions. In the first case, costs are generated randomly from a Uniform (0, 1) distribution, and then the values are multiplied by 100. In the second case, unit costs are considered for all street segments. In the third case, costs are generated randomly from a truncated Normal (0, 1) distribution with the left tail truncated at 0 and the right tail truncated at 1, and then the values are multiplied by 100. We use R software version 3.3.1 to generate the data for the computational experiments and Gurobi version 7.0 to solve the models. We ensure that the generated data are feasible for the linear Stage 1 IP and that all entries in the constraint matrix are non-zero (the p_{ij} 's in the real word will be non-zero). The constraint matrix has dimension $|I| \times |E \cup A|$. We use an i7 CPU with 32 GB RAM and a one-hour time limit. If an optimal value is not found within the time limit, the best feasible solution value (denoted by V) and the best available bound (denoted by B) are reported.

In Table 3, we show the comparison of the performance for the linear Stage 1 IP for the three cost structures with respect to the running times (in seconds) and the optimality gap ($100(\frac{V-B}{B})$). The value of the specified likelihood L_i is set at 0.95 for all i . For each pair of $|I|$

Table 3: Comparison of the linear Stage 1 IP performance for the three different cost structures.

$ I $	$ E \cup A $	Uniform Cost		Unit Cost		Normal Cost	
		Time	Gap	Time	Gap	Time	Gap
20	10	0.000	Optimal	0.016	Optimal	0.016	Optimal
10	20	0.047	Optimal	0.016	Optimal	0.031	Optimal
100	20	0.190	Optimal	0.105	Optimal	0.111	Optimal
20	100	0.031	Optimal	0.081	Optimal	0.043	Optimal
200	100	0.224	Optimal	68.857	Optimal	0.237	Optimal
100	200	0.172	Optimal	56.458	Optimal	0.158	Optimal
1000	200	2.359	Optimal	3600.025	60%	2.269	Optimal
200	1000	0.688	Optimal	3600.026	50%	0.762	Optimal
2000	1000	10.327	Optimal	3600.063	125%	10.564	Optimal
1000	2000	6.102	Optimal	3600.249	100%	6.201	Optimal

Table 4: Comparison of the linear Stage 1 IP objective value for specified likelihood values of 0.95 and 0.75 for all meters.

$ I $	$ E \cup A $	Uniform Cost		Unit Cost		Normal Cost	
		0.95	0.75	0.95	0.75	0.95	0.75
20	10	210.3068	90.1101	5	3	185.8380	77.9464
10	20	90.9012	44.0025	4	2	78.6606	37.7965
100	20	185.6485	101.2878	6	4	161.9885	88.9582
20	100	42.0435	23.7557	4	3	36.0238	20.3450
200	100	60.6152	27.9134	6	4	51.9349	23.9070
100	200	31.1048	16.9587	5	3	26.6314	14.5172
1000	200	50.0808	23.3395	B = 5, V = 8	B = 3, V = 5	42.8871	19.9795
200	1000	5.0900	2.7574	B = 4, V = 6	B = 3, V = 4	4.3551	2.3593
2000	1000	7.4823	3.8587	B = 4, V = 9	B = 2, V = 6	6.4021	3.3016
1000	2000	1.9747	0.9722	B = 4, V = 8	B = 2, V = 5	1.6896	0.8319

and $|E \cup A|$, the linear Stage 1 IP had similar performance for both the uniform cost structure and the normal cost structure in terms of the running times and the optimality gap. All the models are solved to optimality with the running times ranging from one-hundredth of a second for $|I| \times |E \cup A|$ value of 200 to 10 seconds for $|I| \times |E \cup A|$ value of 2,000,000. For each pair of $|I|$ and $|E \cup A|$, the running times are significantly larger for the unit cost structure compared to the other two cost structures. This is probably due to the inherent symmetry in the linear Stage 1 IP for the unit cost structure, since each street segment has the same weight in the objective function. The models are not solved to optimality within the one-hour time limit for $|I| \times |E \cup A|$ values greater than 20,000. The optimality gap is more than 50% and more than 100% for $|I| \times |E \cup A|$ value of 200,000 and 2,000,000, respectively.

In Table 4, for each of the three cost structures we show the comparison of the objective

value for the linear Stage 1 IP for L_i values of 0.95 and 0.75 for all i . If the linear Stage 1 IP is solved to optimality, the objective value is smaller for L_i values of 0.75 compared to 0.95 for each pair of $|I|$ and $|E \cup A|$ and for each of the three cost structures. For the unit cost structure and for $|I| \times |E \cup A|$ values greater than 20,000, the interval $[B, V]$ is tighter and shifted towards zero for L_i values of 0.75 compared to 0.95. This demonstrates the fact that the smaller the specified likelihood values for reading meters, the smaller is the Stage 1 IP objective value.

The Stage 1 IP selects the street segments with the lowest total cost (length) that guarantee the specified likelihood of reading the meters. The Stage 2 IP adds deadhead segments with the lowest total cost (length) to complete the full route, starting and ending at the depot. It might be the case that the street segments selected by the Stage 1 IP are not the best selections for the full route. The street segments selected by the Stage 1 IP might be farther away from each other. This may lead to a larger total length for the full route. To help prevent this, an alternate objective function for the Stage 1 IP may be minimizing the total number of street segments (unit cost structure), i.e., $c_j = 1$ for all j . For the unit cost structure, our computational experiments showed that the linear Stage 1 IP with a one-hour running time had an optimality gap of more than 100% for the $|I| \times |E \cup A|$ value of 2,000,000. Since the $|I| \times |E \cup A|$ value for our data set ($6,067 \times 1,575$) is an order of magnitude larger, it is not possible to arrive at Stage 1 IP solutions with smaller optimality gaps in a reasonable amount of time. So, rather than using the unit cost structure as the only objective function in the Stage 1 IP, it might be useful to explore the option of having the unit cost structure added to the Stage 1 IP as the second objective function. We compare the routes generated using the single-objective Stage 1 IP (discussed in Section 3) and the bi-objective Stage 1 IP in the simulation experiments conducted in Section 7 with our data set. The bi-objective Stage 1 IP is given by the following.

$$\text{(Bi-objective Stage 1 IP)} \quad \min \quad \sum_{j \in E \cup A} c_j x_j \quad (11)$$

$$\min \quad \sum_{j \in E \cup A} x_j \quad (12)$$

$$\text{s.t.} \quad \prod_{j \in E \cup A} (1 - p_{ij})^{x_j} \leq (1 - L_i) \quad \forall i \in I \quad (13)$$

$$x_j \in \{0, 1\} \quad \forall j \in E \cup A \quad (14)$$

The objective function (11) is the same as the objective function (1) in the single-objective Stage 1 IP. The objective function (12) minimizes the total number of street segments selected. Constraints (13) and (14) are the same as constraints (2) and (3), respectively, in the single-objective Stage 1 IP. A lexicographic approach (establishing a pre-defined ordering between the competing objective functions) is used to solve the bi-objective Stage 1 IP. First, the single-objective Stage 1 IP is solved with the objective function (11). Then, the value of the objective function (12) is improved without allowing the value of the objective function (11) to increase.

5 Heuristics for the Stage 2 IP

Frederickson et al. (1978) showed that the mixed rural postman problem is NP-complete. A mixed rural postman problem with a particular node (depot) that is a part of the route is also NP-complete. This is the problem we solve in the Stage 2 IP. Therefore, for any realistic large data sets, similar to the size of our data set (6,067 meters, 1,575 street segments, and 1,072 nodes), any exact method of solving the Stage 2 IP potentially has long running times with solutions that have large optimality gaps. We discuss a fast metaheuristic to generate near-optimal solutions in a short amount of time.

5.1 Route Generator

Given a set of required street segments, the goal is to find the shortest route for the meter reading vehicle that starts and ends at the depot and traverses all required street segments. By assuming that the vehicle always takes the shortest path between any two required street segments, the aim of the route generator is reduced to finding an optimal permutation of all required street segments and, in the case of edges, the direction in which they should be traversed. This shortest path between each pair of nodes is computed using Dijkstra’s algorithm (Dijkstra 1959).

The route generator has two phases, the constructive phase and the improvement phase. During the constructive phase, the route starts from the depot node and, based on the nearest-neighbor method, consecutively visits the closest required street segments. Since the edges can be traversed in both directions, the node (among the two nodes representing an edge) closest to the preceding node in the route is visited first. After a complete initial route is constructed, the

Table 5: Local search operators in the variable neighborhood descent metaheuristic.

Reverse	Change the direction in which a required edge is traversed by the vehicle. Required arcs remain unaffected.
Relocate	Remove a required street segment from the current solution and insert it at a different location in the route.
2-opt	Reverse the order in which the vehicle visits the street segments in a subsequence of required street segments.

improvement phase improves the solution using three different local search operators embedded in a variable neighborhood descent metaheuristic. The three local search operators are briefly described in Table 5. The variable neighborhood descent metaheuristic framework is considered very effective for solving routing problems (Defryn and Sörensen 2017, Hansen et al. 2017, Wassan et al. 2017).

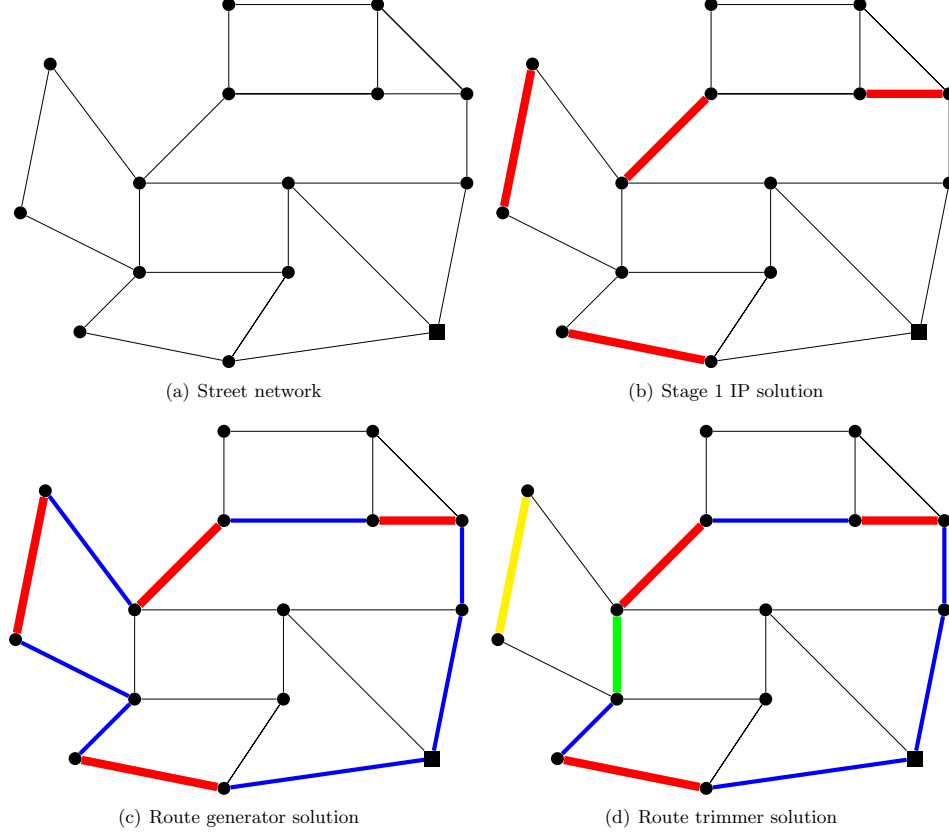
5.2 Route Trimmer

The full route obtained from the route generator contains deadhead street segments in addition to the required street segments. This leads to the fact that the probabilities of successfully reading the meters are more than the specified likelihoods (L_i), which is already guaranteed by the required street segments. Although this will further reduce the chance of missing meters, it comes at a cost of increased route length. To account for this, the route trimmer aims at decreasing the total route length while still assuring feasibility, i.e., the probability of successfully reading each meter i from the full route is at least the specified likelihood L_i . The route trimmer makes use of a remove and repair procedure to decrease the number of required street segments and, therefore, the total route length. In the Appendix, we give the algorithm for the remove and repair procedure of the route trimmer in Table 10.

When the remove and repair procedure terminates, a second procedure of the route trimmer looks for remaining redundancies in the list of required street segments for the current route. When a required street segment lies on the shortest path between its predecessor and successor required street segments, that particular required street segment would be visited by the vehicle as a deadhead segment. Therefore, it can be removed from the list of the required street segments without affecting feasibility or the total route length. This simplifies the representation of the vehicle route and speeds up the simulation experiments conducted in Section 7.

The route generator is used extensively in the route trimming procedure. This strengthens

Figure 2: (Color online) The route generator and the route trimmer applied to a small example. Red lines denote the required street segments. Blue lines denote the deadhead segments. Yellow line denotes the required street segment that is removed. Green line denotes the new street segment added to the route as a replacement for the yellow line.



our motivation for choosing a fast metaheuristic for the route generator. To further speed up the route generator and to avoid solving the same instance multiple times during our simulation experiments, a pool of solutions is maintained. Before using the route generator, this pool is checked for instances that have been solved already.

In Figure 2, we show how the route generator and the route trimmer work for a small example. Figure 2a shows the street network. The route of the vehicle starts from and ends at the depot denoted by the black square. The remaining nodes in the network are denoted by black dots. The black lines are the edges in the network (assume that there are no arcs). Figure 2b shows the Stage 1 IP solution. The red lines are the required street segments that the meter reading vehicle needs to traverse to guarantee the specified service levels (L_i). Figure 2c shows the route produced by the route generator. The blue lines are the deadhead segments added by the route generator to connect the required street segments in the shortest possible

manner. Figure 2d shows the route produced by the route trimmer. The route trimmer finds the required street segment that has the largest marginal cost in the current route (Figure 2c), denoted by the yellow line, and replaces it with another street segment, denoted by the green line, such the new route (Figure 2d) has a smaller length and the specified service levels (L_i) are still satisfied.

6 Bayesian Updating

In order to solve the Stage 1 IP, we need to estimate the values of the probabilities (p_{ij}). Every time the meter reading vehicle collects readings, it adds more data to the previous readings. With more data, we expect that the estimates of the p_{ij} 's will be more accurate. Therefore, the routes generated by the two-stage IP will be of higher quality. They will be better at capturing the uncertain signals thereby further reducing the number of missed reads.

There are some serious issues if we use regression to estimate the p_{ij} 's at every time period with the new data. Suppose in time period 1 we observe the first set of i.i.d. data denoted by y_1 . We run the regression on y_1 . In time period 2, we observe a second set of i.i.d. data denoted by y_2 , independent of y_1 . We run the regression on y_1 and y_2 together as a single data set, and so on. We are regressing on the older data sets repeatedly which makes this process of estimation inefficient. Data sets from different time periods are given equal weights in the regression. In practice, utility companies need to use different weights for the data based on seasonality and other factors. For example, during the summer season, meter reading data from the previous summer is more important and accurate compared to the meter reading data from the previous winter. Also, new obstacles may appear between a meter and a street segment, new meters may appear in the vicinity of a meter causing more interference, and a decrease in the battery level of an RFID tag will reduce the signal range of the meter. All these factors make the most recent meter reading data more accurate. Therefore, utility companies should be able to apply different weights to parts of the data accordingly. If we estimate the p_{ij} 's at every time period with new data, i.e., every time the meter reading vehicle collects data from traversing a route, using concepts from Bayesian statistics, then we can avoid the two drawbacks faced while updating using regression.

Bayesian updating works as follows:

$$\begin{aligned}
\pi(\beta|y_1) &\propto p(\beta)L(\beta|y_1) \\
\pi(\beta|y_1, y_2) &\propto \pi(\beta|y_1)L(\beta|y_2) \\
&\vdots \\
\pi(\beta|y_1, \dots, y_T) &\propto \pi(\beta|y_1, \dots, y_{T-1})L(\beta|y_T) = p(\beta)L(\beta|y_1, \dots, y_T).
\end{aligned}$$

In time period 1, $p(\beta)$ is the prior information on the unknown parameter vector β , and $L(\beta|y_1)$ is the likelihood function obtained from i.i.d. data y_1 . The posterior distribution is $\pi(\beta|y_1)$. In time period 2, the prior is $\pi(\beta|y_1)$, which is the posterior from time period 1, and so on. This process can be repeated and the model will continue to update the posterior distributions as we collect new data. In time period T , the posterior is $\pi(\beta|y_1, \dots, y_T)$, which does not depend on the sequence in which data arrive. This is exactly the same result that would have obtained if all the i.i.d. data (y_1, \dots, y_T) had been gathered at the same time because $L(\beta|y_1, \dots, y_T) = \prod_{i=1}^n L(\beta|y_i)$. Bayesian updating is much faster than regression since analysis is done only on the new incoming data at each time period. Data from different time periods can be weighted differently in Bayesian updating depending on the requirements of the utility companies. The idea is to re-solve the two stage IP at the end of each time period with the new posterior distribution of the probabilities (p_{ij}) that is obtained and, thereby, have an iterative algorithm to generate more robust routes at the end of each time period. The unknown parameters in the Bayesian models are estimated using Markov Chain Monte Carlo (MCMC) simulations.

The dependent variable in the data is denoted by Read_OR_Not_{ij} (whether or not meter i was read from street segment j). The data elements have the form of 1 and 0, where 1 indicates that meter i is read from street segment j and 0 indicates that meter i is not read from street segment j . The predicted values of the dependent variable have to be between 0 and 1 which will denote the probabilities (p_{ij}) . Based on the type of the data we have and our requirements on the predicted values of the dependent variable, logit and probit models are considered. The independent variables are: $\text{Shortest_Distance}_{ij}$ (shortest distance between meter i and street segment j), No_of_Pulses_j (number of pulses the meter reading vehicle can receive from the meter while traveling on street segment j), and No_of_Customers_i (number of meters within

500 feet from meter i ; 500 feet is the range of the RFID signals as specified by ITRON, so the signals are strong enough to interfere with each other within 500 feet). $\text{Shortest_Distance}_{ij}$ should have a negative coefficient because the larger the shortest distance between meter i and street segment j is, the smaller the value of p_{ij} . No_of_Pulses_j is obtained from the amount of time the meter reading vehicle spent on street segment j divided by the time interval between the RFID signal transmissions. If the meter reading vehicle travels at a higher speed through street segment j , then the time spent by the vehicle on street segment j is smaller and, therefore, the No_of_Pulses_j is lower. No_of_Pulses_j should have a positive coefficient because the greater the number of pulses the meter reading vehicle can receive from the meters while traveling on street segment j , the larger the value of p_{ij} . No_of_Customers_i is a measure of the density of meters in a region. It is important because, with a large number of meters in a region, the interference of the RFID signals is greater, so the signals die out quickly. No_of_Customers_i should have a negative coefficient because the greater the number of meters surrounding meter i , the smaller the value of p_{ij} .

We use only the traversed street segments to estimate the parameters of the models. We can only determine that a meter was read or not read on street segments traversed by the meter reading vehicle. Thus, the data used for estimating the parameters of the models has 6,067 meters and 829 street segments (counting repetitions).

6.1 Logit Model and Probit Model

The logit and probit models are given by

$$g(p_{ij}) = \hat{\beta}_1 + \hat{\beta}_2 \times \text{Shortest_Distance}_{ij} + \hat{\beta}_3 \times \text{No_of_Pulses}_j + \hat{\beta}_4 \times \text{No_of_Customers}_i$$

where g is the link function, $p_{ij} = \mathbb{E}(\text{Read_OR_Not}_{ij})$, $\hat{\beta}_k = \mathbb{E}(\beta_k)$, and $\mathbb{E}()$ denotes the expected value. For the logit model, $g(p_{ij}) = \ln(p_{ij}/1 - p_{ij})$, and for the probit model $g(p_{ij}) = \Phi^{-1}(p_{ij})$, where Φ is the cumulative Normal $(0, 1)$ distribution function. We have to estimate the unknown parameter vector $\beta = (\beta_1, \beta_2, \beta_3, \beta_4)^\top$ which is a 4-dimensional vector of coefficients. The data matrix X is $(N \times 4)$ -dimensional, where N is the size of the data set, and the entries in the first column of X are 1's. X_k denotes row k of X . In time period 1, we do not have any prior information about β . We rely on the information obtained from the data. Therefore, the prior

is set to a vague prior, i.e., the prior will have minimal effect on the posterior distribution of time period 1.

Both models have their pros and cons. Error terms in logit models have a logistic distribution, whereas error terms in probit models have a normal distribution. The logistic distribution has heavier tails compared to the normal distribution, so logit models are more robust than probit models. Logit models have a better fit to data that are more spread out in the tails. The normal distribution is the conjugate prior for the likelihood function in probit models. Therefore, the unknown parameters in the probit model can be estimated using an exact algorithm. However, neither the normal distribution nor any other distribution from the exponential family is the conjugate prior for the likelihood function in logit models. Therefore, the unknown parameters in the logit model have to be estimated using a non-exact algorithm.

The Metropolis-Hastings (MH) Random Walk algorithm (Metropolis et al. 1953, Hastings 1970) is used to estimate the parameters of the logit model and has four steps.

Step 1. Choose a starting value for β .

Step 2. The random walk chain $\beta^{new} = \beta^{old} + \epsilon$ where $\epsilon \sim \text{Multivariate Normal}(0, s^2 H^{-1})$ generates candidate realizations for β , $s = 2.3/\sqrt{\text{dimension}(\beta)}$ (Marin and Robert 2014, Press 2003), and H is the Hessian of the log likelihood function for the logit model.

Step 3. Accept β^{new} with probability $\alpha = \min\{1, \frac{\pi(\beta^{new}|y, X)}{\pi(\beta^{old}|y, X)}\}$.

Step 4. Repeat Steps 2 and 3.

The Gibbs sampling algorithm (Albert and Chib 1993) is used to estimate the parameters of the probit model. The setup for this algorithm is as follows. The prior distribution is $p(\beta) = \text{Multivariate Normal}(B_0, V_0)$. We have $y_k^* \sim \text{Normal}(X_k \beta, 1)$ and $y_k = \text{Indicator}(y_k^* > 0)$, where y is the dependent variable and y^* is the latent variable. The distribution of y^* is given by $p(y_k^*|y_k = 0, \beta) = \text{Normal}(X_k \beta, 1) \times \text{Indicator}(y_k^* \leq 0)$ and $p(y_k^*|y_k = 1, \beta) = \text{Normal}(X_k \beta, 1) \times \text{Indicator}(y_k^* > 0)$. Therefore, the posterior distribution is $\pi(\beta|y^*) = \text{Multivariate Normal}((X^\top X + V_0^{-1})^{-1}(X^\top y^* + V_0^{-1}B_0), (X^\top X + V_0^{-1})^{-1})$. The Gibbs sampling algorithm for the probit model has four steps.

Step 1. Choose a starting value for β .

Step 2. Draw $[y^*|y, \beta]$.

Step 3. Draw $[\beta|y^*]$.

Step 4. Repeat Steps 2 and 3.

6.2 Logit Model and Probit Model Results

The size of the data set used for estimating the parameters of the logit model and the probit model is $N = 6,067 \times 829$ (~ 5 million). To verify that our choice of the prior distribution on the unknown parameters for both the logit model and the probit model does not have much effect on the posterior distribution, we also estimate the parameters of both the models using regression. If the parameter values for the regression and the Bayesian estimation match, this indicates that our choice of the prior for the Bayesian estimation fulfills our requirement of a vague prior. In subsequent time periods, when new data are gathered, the parameters can be updated using the Bayesian updating algorithms.

In logistic regression and probit regression, the parameters are estimated using the maximum likelihood estimation (MLE) method. We use McFadden's Adjusted R^2 to assess our models where values between 0.2 and 0.4 indicate similar quality model fit as Ordinary Least Squares Adjusted R^2 values between 0.7 and 0.9 (Domencich and McFadden 1975, Louviere et al. 2000).

In the MH random walk algorithm for the logit model, the prior for β is set to a multivariate normal distribution with the mean vector as the zero vector, the variances are 10,000, and the covariances are zero. The first 5,000 samples are considered as the burn-in period and the next 10,000 samples are collected for analysis. The starting value for β is set to the maximum likelihood estimator for the likelihood function of the logit model. The acceptance rate for the new values of β generated from the Markov chain is around 33% (the acceptance rate should be between 30-35% for an optimal combination of exploration and exploitation steps).

In Table 6, we give the logit model results. The mean and standard deviation of the β_i 's from the MH random walk algorithm and logistic regression are presented. Since for each i , β_i values match for both the logistic regression and the MH random walk algorithm, our choice of the prior in the MH random walk algorithm serves the purpose of a vague prior. The McFadden's Adjusted R^2 value of 0.223 for the logistic regression indicates very good model fit. Also, all coefficients of the regression model are significant at the 1% level, and the signs of the coefficients

Table 6: Logit model results.

Coefficient	Logistic Regression	MH Random Walk Algorithm
Intercept (β_1)	-1.242*** (0.010)	-1.242 (0.010)
Shortest_Distance (β_2)	-0.003*** (0.000009)	-0.003 (0.000009)
No_of_Pulses (β_3)	0.019*** (0.00008)	0.019 (0.00008)
No_of_Customers (β_4)	-0.003*** (0.00005)	-0.003 (0.00005)
McFadden's Adjusted R^2	0.223	

***p<0.01

Table 7: Probit model results.

Coefficient	Probit Regression	Gibbs Sampling Algorithm
Intercept (β_1)	-1.000*** (0.004)	-1.024 (0.091)
Shortest_Distance (β_2)	-0.001*** (0.000004)	-0.001 (0.000105)
No_of_Pulses (β_3)	0.007*** (0.00003)	0.007 (0.00069)
No_of_Customers (β_4)	-0.002*** (0.00002)	-0.002 (0.00011)
McFadden's Adjusted R^2	0.200	

***p<0.01

of the three independent variables are what we expected. This indicates that we can perform Bayesian updating for the logit model after receiving new data points instead of using logistic regression.

In the Gibbs sampling algorithm for the probit model, in order to set the prior for β , B_0 is set to the zero vector and V_0 is set to the diagonal matrix with diagonal entries of 10,000. We collected 10,000 samples for analysis. The starting value for β is set to the zero vector.

In Table 7, we give the probit model results. The mean and standard deviation of the β_i 's from the Gibbs sampling algorithm and probit regression are presented. Since for each i , β_i values match for both the probit regression and the Gibbs sampling algorithm, our choice of the prior in the Gibbs sampling algorithm serves the purpose of a vague prior. The McFadden's

Adjusted R^2 value of 0.200 for the probit regression indicates very good model fit. Also, all of the coefficients of the regression model are significant at the 1% level, and the signs of the coefficients of the three independent variables are what we expected. This indicates that we can perform Bayesian updating for the probit model after receiving new data points instead of using probit regression.

6.3 Hierarchical Probit Model

For our meter reading problem, hierarchical models consider the signal transmission behavior of individual meters and their interactions with the signals from other meters. Bayesian updating for the hierarchical probit model is a more complex method for estimating the p_{ij} 's but the estimates are more accurate compared to Bayesian updating for the logit model and the probit model. The hierarchical probit model accounts for the uncertain behavior of each meter separately while also accounting for the similarity between meters.

The rationale behind using a hierarchical model for updating the probability estimates is that each meter is inherently different from every other meter. Some meters are surrounded by physical obstacles, some are on elevated ground, and some are old. New meters have better technology. The meters have different stages of battery life. As the battery level of the meters drop below a certain threshold, the signal transmission range decreases. All of these factors affect the signal transmission behavior of the meters.

Let n be the number of meters and m be the number of street segments (counting repetitions) traversed by the meter reading vehicle. Group the observations Read_OR_Not_{ij} into n buckets, where bucket i contains observations on meter i . Each bucket contains m observations with one from each traversed street segment. The probit model for each group i is called the lower level model for meter i and is given by

$$\Phi^{-1}(p_{ij}) = \hat{\beta}_{i,1} + \hat{\beta}_{i,2} \times \text{Shortest_Distance}_{ij} + \hat{\beta}_{i,3} \times \text{No_of_Pulses}_j.$$

where $\hat{\beta}_{i,k} = \mathbb{E}(\beta_{i,k})$ and $\mathbb{E}()$ denotes the expected value. The lower level unknown parameter vector $\beta_i = (\beta_{i,1}, \beta_{i,2}, \beta_{i,3})^\top$ for each group i are 3-dimensional vector of coefficients and are used as dependent variables in a multivariate linear model called the higher level model. The data matrices X_i for each group i are $(m \times 3)$ -dimensional and the entries in the first

column of each matrix are 1's. The multivariate linear model is $B = Z\Theta + \Delta$, where Δ is the error term. $B = (\beta_1^\top, \dots, \beta_n^\top)^\top$ is an $(n \times 3)$ -dimensional matrix of the dependent variables. $Z = (z_1^\top, \dots, z_n^\top)^\top$ is an $(n \times 2)$ -dimensional matrix of the independent variables, where $z_i = (1, \text{No_of_Customers}_i)^\top$ for each i is a 2-dimensional vector. The higher level unknown parameter matrix $\Theta = \begin{pmatrix} \theta_{1,1} & \theta_{2,1} & \theta_{3,1} \\ \theta_{1,2} & \theta_{2,2} & \theta_{3,2} \end{pmatrix}$ is a (2×3) -dimensional matrix of coefficients from the multivariate linear model. For each i , the multivariate linear model can be written as $\beta_i = \Theta^\top z_i + \delta_i$, where $\delta_i \sim \text{Multivariate Normal}(0, \Lambda)$ is the error term for the group i . We have to estimate the lower level parameters β_i 's for each meter i and the higher level parameters Θ and Λ . In time period 1, we do not have any prior information about the parameters, so we rely on the information obtained from the data. Therefore, the priors are set to vague priors.

The Gibbs sampling algorithm is used to estimate the parameters of the hierarchical probit model. The setup for this algorithm is as follows. The prior distributions for the lower level parameters are $p(\beta_i) = \text{Multivariate Normal}(\Theta^\top z_i, \Lambda)$ for each i . The prior distributions for the higher level parameters are $p(\text{vec}[\Theta^\top]) = \text{Multivariate Normal}(M_0, V_0)$ and $p(\Lambda) = \text{Inverse-Wishart}(c_0, D_0)$, where vec denotes the operator that transforms a matrix into a vector by concatenating columns. We have $y_i^* \sim \text{Multivariate Normal}(X_i \beta_i, 1)$ and $y_{ij} = \text{Indicator}(y_{ij}^* > 0)$, where $y_i = (y_{i1}, \dots, y_{im})^\top$ is the dependent variable vector and $y_i^* = (y_{i1}^*, \dots, y_{im}^*)^\top$ is the latent variable vector for group i . Therefore, the posterior distributions for the lower level parameters are $\pi(\beta_i | y_i^*, \Theta, \Lambda) = \text{Multivariate Normal}((X_i^\top X_i + \Lambda^{-1})^{-1}(X_i^\top y_i^* + \Lambda^{-1} \Theta^\top z_i), (X_i^\top X_i + \Lambda^{-1})^{-1})$ for each i , and the posterior distributions for the higher level parameters are $\pi(\text{vec}[\Theta^\top] | \{y_i^*\}, \Lambda, \{\beta_i\}) = \text{Multivariate Normal}((Z^\top Z \otimes \Lambda^{-1} + V_0^{-1})^{-1}((Z^\top \otimes \Lambda^{-1})\text{vec}[B^\top] + V_0^{-1}M_0), (Z^\top Z \otimes \Lambda^{-1} + V_0^{-1})^{-1})$ and $\pi(\Lambda | \{y_i^*\}, \Theta, \{\beta_i\}) = \text{Inverse-Wishart}(c_0 + n, D_0 + (B - Z\Theta)^\top (B - Z\Theta))$. The Gibbs sampling algorithm for the hierarchical probit model has six steps.

Step 1. Choose starting values for β_i 's, Θ , and Λ .

Step 2. Draw $[y_i^* | y_i, \beta_i]$ for each group i .

Step 3. Draw $[\beta_i | y_i^*, \Theta, \Lambda]$ for each group i .

Step 4. Draw $[\Theta | \{y_i^*\}, \Lambda, \{\beta_i\}]$.

Table 8: Hierarchical probit model results for the higher level parameter matrix.

Coefficient	Gibbs Sampling Algorithm
$\theta_{1,1}$	-0.890 (0.241)
$\theta_{2,1}$	-0.002 (0.0009)
$\theta_{3,1}$	0.004 (0.002)
$\theta_{1,2}$	-0.0002 (0.0001)
$\theta_{2,2}$	-0.000003 (0.000004)
$\theta_{3,2}$	0.0000006 (0.000005)

Step 5. Draw $[\Lambda|\{y_i^*\}, \Theta, \{\beta_i\}]$.

Step 6. Repeat Steps 2 to 5.

6.4 Hierarchical Probit Model Results

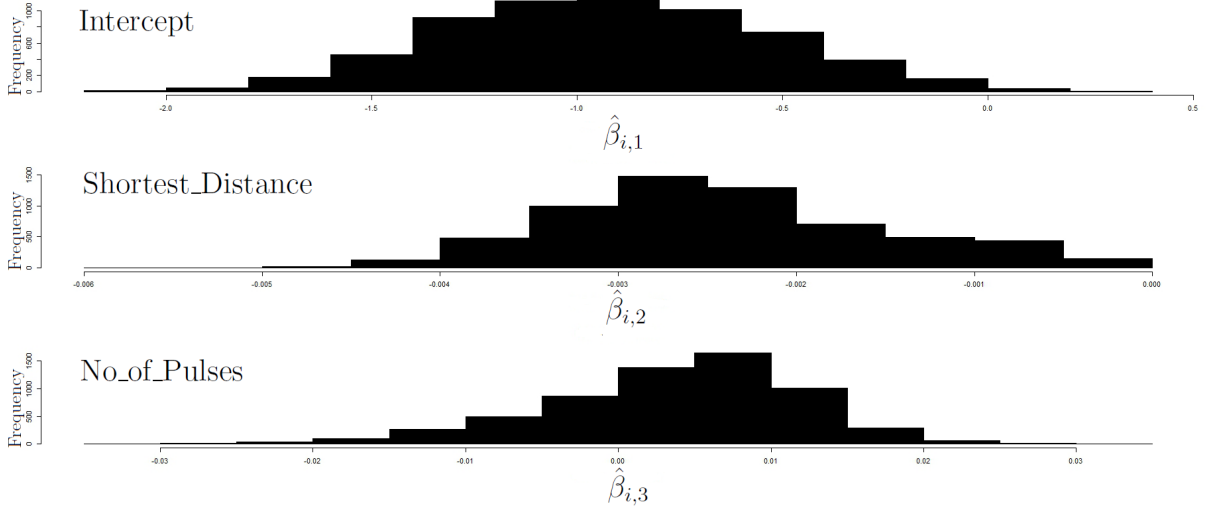
The size of the data sets used for estimating the lower level parameters in the probit models for each meter i is $m = 829$. The size of the data set used for estimating the higher level parameters in the multivariate linear model is $n = 6,067$.

In the Gibbs sampling algorithm for the hierarchical probit model, in order to set the priors for β_i 's, Θ , and Λ , M_0 is set to the zero vector and V_0 is set to the diagonal matrix with diagonal entries of 1,000. We set c_0 to seven and D_0 to the diagonal matrix with diagonal entries of three (Press 2003, Rossi et al. 2005). We collected 10,000 samples for analysis. The starting values for the β_i 's and the starting value for Θ are set to the zero vector. The starting value for Λ is sampled from Inverse-Wishart ($c_0 + n, D_0$).

In Table 8, we give the hierarchical probit model results for Θ . The mean and standard deviation of the $\theta_{i,j}$'s from the Gibbs sampling algorithm are presented. Using these values, the β_i 's are calculated for each meter i .

In Figure 3, we show the histograms of the means of the β_i 's from the Gibbs sampling

Figure 3: Histograms of the means of the lower level parameters from the Gibbs sampling algorithm for the hierarchical probit model.



algorithm for the hierarchical probit model. The three histograms belong to the coefficients of the Intercept ($\hat{\beta}_{i,1}$), the Shortest_Distance ($\hat{\beta}_{i,2}$), and the No_of_Pulses ($\hat{\beta}_{i,3}$), respectively, for the lower level probit model in the hierarchical probit model. The histograms show the variation in the coefficients for different meters that are not captured in the logit model or the probit model. All of the meters have the same equation for estimating the p_{ij} 's for the logit model and the probit model. The hierarchical probit model gives individualized probability predictions p_{ij} for each meter i . Thus, for the hierarchical probit model, each meter has its unique equation for estimating the p_{ij} 's.

7 Simulation Experiments

To compare and quantify the performance of the three Bayesian updating models on route quality, simulation experiments are conducted using the data set with 6,067 meters, 1,575 street segments, and 1,072 nodes. As the meter reading vehicle makes more trips and collects more data, the parameters of the Bayesian updating models should get closer to the actual parameter values. The actual parameter values depend on the street network and the distribution of the meters in the street network. This will be demonstrated by the fact that the vehicle routes will be adjusted over time to reduce the number of missed meters and still being cost-effective.

7.1 Actual Reading Probabilities

To calculate the probabilities p_{ij} 's for the three Bayesian updating models, we use the parameter values estimated in Section 6. The p_{ij} 's obtained are considered to be the actual probabilities, denoted by \bar{p}_{ij} 's, with which meter i is read from street segment j at least once. This assumption is reasonable since the size of the data set on which the model parameters are estimated is very large ($N \sim 5$ million for the logit model and the probit model, and $m = 829$ and $n = 6,067$ for the hierarchical probit model). To determine whether or not meter i is successfully read from street segment j , we consider a binomial random variable $Y_{ij} \sim \text{Binomial}(M, \bar{p}_{ij})$, where M is the number of times street segment j is traversed in the route. Meter i is considered missed from the route of the vehicle if $Y_{ij} = 0$ for all street segments j in the route, i.e., meter i has not been read from any of the street segments traversed by the vehicle.

For the logit model, $\ln(\bar{p}_{ij}/1 - \bar{p}_{ij}) = -1.242 - 0.003 \times \text{Shortest_Distance}_{ij} + 0.019 \times \text{No_of_Pulses}_j - 0.003 \times \text{No_of_Customers}_i$, where $\text{Shortest_Distance}_{ij}$ is in meters.

For the probit model, $\Phi^{-1}(\bar{p}_{ij}) = -1.024 - 0.001 \times \text{Shortest_Distance}_{ij} + 0.007 \times \text{No_of_Pulses}_j - 0.002 \times \text{No_of_Customers}_i$, where $\text{Shortest_Distance}_{ij}$ is in meters.

For the hierarchical probit model, $\mathbb{E}(\beta_i) = \mathbb{E}(\Theta^T)z_i$ for each i . Therefore,

$$\begin{pmatrix} \hat{\beta}_{i,1} \\ \hat{\beta}_{i,2} \\ \hat{\beta}_{i,3} \end{pmatrix} = \begin{pmatrix} -0.890 & -0.0002 \\ -0.002 & -0.000003 \\ 0.004 & 0.0000006 \end{pmatrix} \begin{pmatrix} 1 \\ \text{No_of_Customers}_i \end{pmatrix}.$$

So, $\Phi^{-1}(\bar{p}_{ij}) = (-0.890 - 0.0002 \times \text{No_of_Customers}_i) + (-0.002 - 0.000003 \times \text{No_of_Customers}_i) \times \text{Shortest_Distance}_{ij} + (0.004 + 0.0000006 \times \text{No_of_Customers}_i) \times \text{No_of_Pulses}_j$, where $\text{Shortest_Distance}_{ij}$ is in meters.

7.2 Simulation Model Overview

A simulation starts from an initial vehicle route. This is iteration zero or the initialization step. To construct the initial route, p_{ij} is set to 1 if meter i is within 500 feet from street segment j and 0 otherwise. First, the linear Stage 1 IP is solved to obtain the required street segments. The specified likelihood (L_i) values do not affect the Stage 1 IP solution in iteration zero because of the particular choice of the p_{ij} 's. Then the full route is produced using the route

generator and the route trimmer. This initial route is a deterministic CEVRP solution which is currently used by utility companies. Therefore, we use this initial route as a benchmark for our experiments. In the first iteration, the Y_{ij} 's are generated based on the initial route. Since we do not have any information about the p_{ij} 's during the first iteration (this is analogous to time period 1 as described in Section 6), the prior distributions for the parameters of the Bayesian updating models are set to the vague priors. Using the priors and the meter reading data (Y_{ij} as the dependent variable and $\text{Shortest_Distance}_{ij}$, No_of_Pulses_j , and No_of_Customers_i as the independent variables), we obtain the posterior distributions of the parameters, and thereby the updated p_{ij} 's. A new route is produced based on the updated p_{ij} 's. For all subsequent iterations, the Y_{ij} 's are generated based on the current route; the posterior distributions from the previous iteration are used as the prior distributions.

An iteration in the simulation experiment represents the generation of a new route for the next meter reading day after updating the probabilities p_{ij} 's using the previous meter reading data. Therefore, our simulation model can be used by a utility company as a decision-support tool to generate robust and cost-effective routes. The only difference would be that the utility companies have access to the actual dependent variable values (Read_OR_Not_{ij}) after the vehicle has traversed the route, whereas, for the simulations, we generate the Y_{ij} 's based on the \bar{p}_{ij} 's and the route to identify the missed meters.

7.3 Generating the Network

Our data set is from Hartford, Connecticut. The OAR Bench software (Lum et al. 2018) is used to extract the metadata of the street network (information about nodes, edges, and arcs). The coordinate system used by OAR Bench is the World Geodetic System (WGS) 84. The meter locations in the ArcGIS data are also in WGS 84 format. WGS 84 is the reference coordinate system used by the Global Positioning System (GPS). The Universal Transverse Mercator (UTM) conformal projection (map projections that preserve angles locally) uses a 2-dimensional Cartesian coordinate system to give locations on the surface of the Earth by dividing the Earth into sixty zones. The metadata of the street network and the meter locations are converted from the WGS 84 format to the UTM format using the UTM zone number of Hartford, Connecticut. The street network and the meter locations in the UTM format are considered to be on a flat surface. Therefore, Euclidean geometry can be used to find spatial

Figure 4: (Color online) A view of a portion of the actual street network with meter locations in the UTM format serviced by Connecticut Natural Gas in our data set from Hartford, Connecticut. The red dots represent the meters. This network is used for our simulation experiments.

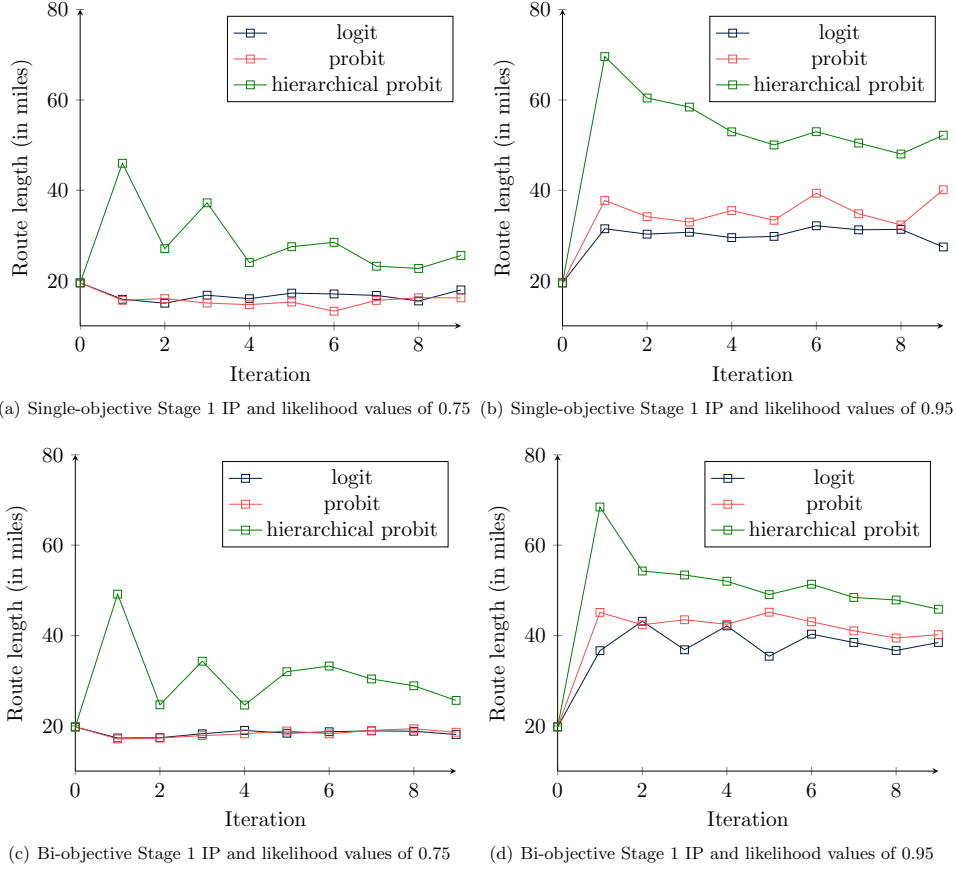


distances between any two points in the network. Figure 4 gives a view of the actual street network with meter locations in the UTM format.

7.4 Simulation Results

Simulations are conducted for each of the three Bayesian updating models, for both the single-objective and the bi-objective linear Stage 1 IP, and for L_i values of 0.95 and 0.75 for all i . In total, we perform 12 ($3 \times 2 \times 2$) simulation experiments. Each simulation is run for nine iterations. The initial route in iteration zero depends on the version of the Stage 1 IP used for that particular simulation experiment. A simulation using the single-objective linear Stage 1 IP

Figure 5: (Color online) Simulation results for the route length.

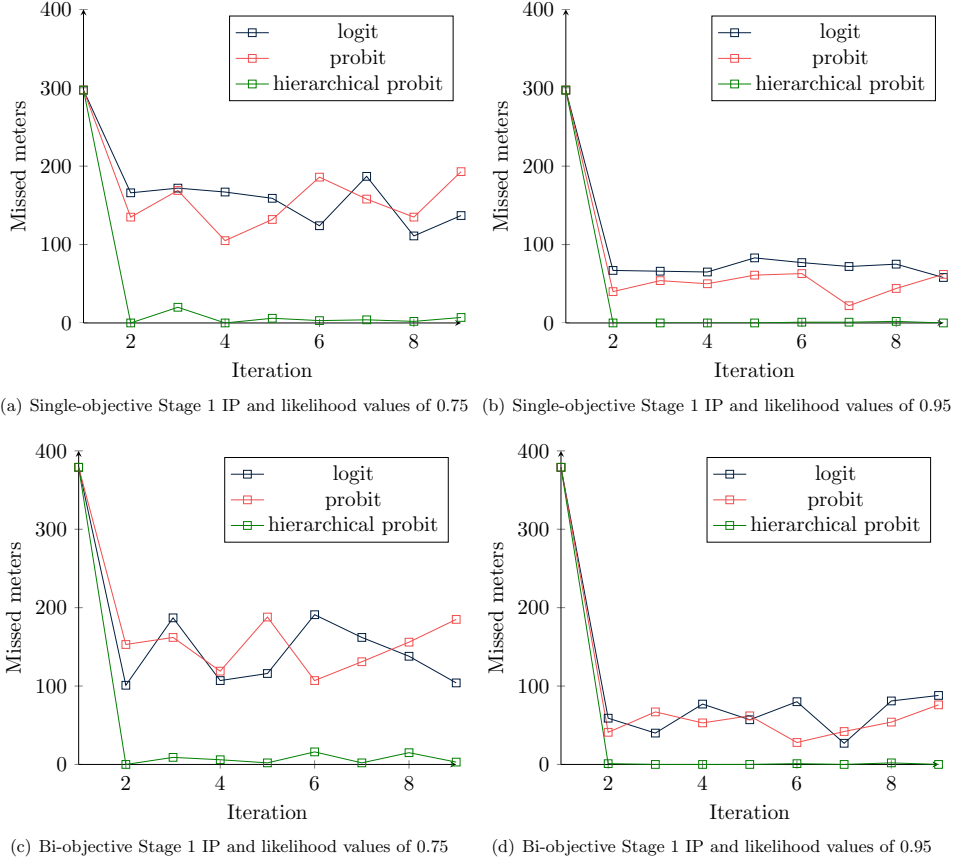


has a different initial route compared to a simulation using the bi-objective linear Stage 1 IP.

The meter reading vehicles are driven at five miles per hour in residential neighborhoods. Drivers are encouraged to drive at a slow speed to increase the chances of reading the uncertain RFID signals. We assume five minutes to manually read a meter (these are the meters that are infeasible with respect to constraints (2) in the Stage 1 IP) after parking the vehicle on the closest street segment. The route lengths are in miles. For each meter that is supposed to be manually read, we add 0.42 miles to the full route length as a proxy for the distance that could have been traversed in five minutes at five miles per hour. We use an i7 CPU with 32 GB RAM for the simulations. We use R software version 3.3.1 to run the Bayesian updating models and Gurobi version 7.5 to solve the linear Stage 1 IP. Each time the linear Stage 1 IP is solved or the route generator is used, a time limit of 10 minutes and two minutes, respectively, is imposed.

In Figures 5 and 6, we show the results of the simulation experiments for the route length and the number of missed meters, respectively. Both figures show the results of the three

Figure 6: (Color online) Simulation results for the number of missed meters.



Bayesian updating models in four different scenarios, namely, single-objective linear Stage 1 IP and likelihood values of 0.75 (scenario a), single-objective linear Stage 1 IP and likelihood values of 0.95 (scenario b), bi-objective linear Stage 1 IP and likelihood values of 0.75 (scenario c), and bi-objective linear Stage 1 IP and likelihood values of 0.95 (scenario d). Figure 5 shows the results from iteration 0 (initial route) through iteration 9, whereas, Figure 6 shows the results from iteration 1 through iteration 9. The missed meters (Figure 6) in iteration k are based on the route (Figure 5) in iteration $k - 1$ and also leads to the formation of the new route in iteration k .

In each of the four scenarios depicted in Figures 5 and 6, the logit model and the probit model do not show any significant differences. The initial routes are approximately 20 miles and there are 279 missed meters in scenarios a and b and 379 missed meters in scenarios c and d. The logit model and the probit model, on average, generated routes 17 miles long and missed 148 meters in scenarios a and c, and routes 37 miles long and missed 60 meters in scenarios b and d. An increase in the likelihood values from 0.75 to 0.95 increased the route length from

17 miles to 37 miles and reduced the number of missed meters from 148 to 60. However, the choice of the Stage 1 IP did not have any substantial impact on the results. For the routes to be operationally impactful to the extent that there is no need to send another vehicle at a later time to read the missed meters, the number of missed meters needs to be reduced further.

The hierarchical probit model, in each of the four scenarios shown in Figure 5, shows a large increase in route lengths compared to the initial routes of 20 miles. However, the route lengths gradually decreased to around 28 miles in scenarios a and c, and around 52 miles in scenarios b and d. The hierarchical probit model, as shown in Figure 6, on average, missed six meters in scenarios a and c and one meter in scenarios b and d. The choice of the Stage 1 IP did not have any significant impact on the results. The probability estimates of the hierarchical probit model get very close to the actual probabilities within a few iterations. Therefore, even with likelihood values of 0.75, the routes generated are very high quality (only six meters are missed from these routes that are just 8 miles longer than the initial routes). These routes should be of high operational and practical relevance for a utility company. Likelihood values of 0.95 make the routes substantially longer, thereby, increasing the cost.

In Bayesian updating, the posterior distribution of the model parameters is a compromise between prior information and the information provided by the new data. This helps in the inference on the parameters since, if the new data are smaller in size (this leads to model parameters with distributions of high variance), we want to rely more on prior knowledge. Conversely, if the data are plentiful and contain high-quality information, then we should not care much about the form of the prior information. The Bayesian updating process automatically considers this trade-off. For the hierarchical probit model, the meter reading data obtained from a route are divided into small segments to learn about the behavior of each meter separately. Therefore, the size of the data is effectively a few orders of magnitude smaller for the hierarchical probit model compared to the logit model and the probit model. This leads to more dependence on the priors for the posterior distributions of the hierarchical probit model. When we estimated the parameters of the Bayesian updating models in Section 6 using our real meter reading data, the logit model and the probit model used $6,067 \times 829$ (~ 5 million) data points, whereas, the hierarchical probit model used 829 data points for each of the lower level probit models and 6,067 data points for the higher level multivariate linear model. The simulation experiments were started using vague priors because we did not have any prior information on the param-

Table 9: Average comparison of the total time to read all the meters.

Model	d_1	h	d_2	Total time
Initial (benchmark) route	20	329	25.58	33.12
Logit or Probit for likelihood values of 0.75	17	148	24.62	17.37
Logit or Probit for likelihood values of 0.95	37	60	18.37	13.62
Hierarchical Probit for likelihood values of 0.75	28	6	8.35	6.65
Hierarchical Probit for likelihood values of 0.95	52	1	6.99	10.95

eters. Therefore, to lower the variance of the posterior distributions, the hierarchical probit model produced longer routes to gather more meter reading data and, thereby, improved the quality of the probability estimates.

For a utility company, the total cost of reading meters in each time period (iteration) is divided into two phases. The first phase is the cost of the CEVRP routes to read the meters automatically. The second phase is the cost of reading the meters that are missed from the first phase route. Typically, a public utility commission does not allow billing cycles to shift more than two or three days. Therefore, it is necessary to send out another vehicle to manually read the missed meters within a few days after the completion of the first phase. We compare the quality of the initial (benchmark) route and the routes generated by the three Bayesian updating models taking into account the cost from both phases. Let d_1 denote the length (in miles) of the route in the first phase. A meter reading vehicle is driven at five miles per hour, so the first phase route will take $d_1/5$ hours to complete. Let h denote the number of meters missed from the first phase route. During the second phase, the missed meters are read manually to ensure success with probability one (a vehicle is driven at a speed of around 15 miles per hour and stops at each missed meter to read it). The second phase is a standard VRP route. A utility company knows which meters are missed from the first phase route. It will have the exact standard VRP route length to read the missed meters. In our simulations, we discuss the results averaged over the nine iterations and need to estimate the route length for the second phase. Kwon et al. (1995) showed that $(0.8326 - 0.0011(h+1) + 1.1147G/(h+1))\sqrt{(h+1)D}$, where D is the area of the rectangular network and G is the ratio of length and breadth of the network such that $G \geq 1$, gives a reasonable estimate of the standard VRP route length as a function of the number of customers (h) on the route. For our data set, $G = 1.5$ and $D = 8.8$ square miles. Let d_2 denote the estimate of the length (in miles) of the route in the second phase; the route will take $d_2/15$ hours to complete. If we assume five minutes to manually read a meter,

it will take $h/12$ hours to read h missed meters. The total time (in hours) required to read all meters is the sum of the first phase time ($d_1/5$) and the second phase time ($d_2/15 + h/12$). In Table 9, we show the average comparison of the total time for the benchmark route and the routes generated by the three Bayesian updating models. All three models took much less time than the benchmark route. For the logit and the probit models, the total time was a few hours less for likelihood values of 0.95 compared to likelihood values of 0.75. This is because the time required to read the missed meters in the second phase was smaller even though the first phase routes were longer for likelihood values of 0.95. The hierarchical probit model does considerably better than the logit or the probit models. For likelihood values of 0.75, the hierarchical probit model takes only 6.5 hours to read all the meters. The hierarchical probit model for likelihood values of 0.95 has a longer first phase route without significantly reducing the second phase time compared to the likelihood values of 0.75.

8 Conclusions and Future Directions

We developed an iterative methodology to read uncertain RFID signals from utility meters using vehicles at some distance. Every time we get access to new meter reading data, we learn about the probabilities p_{ij} 's. The two-stage IP, representing the meter reading problem, is then re-solved to generate routes that are robust for addressing the uncertainty. Even though the routes generated by our procedure are a few miles longer than the benchmark route, the routes are cost-effective when compared to the costs incurred by sending a vehicle at a later time to read the missed meters. The two-stage IP formulation is deterministic even though the meter reading problem has an inherent stochastic set up. The Stage 1 IP, which gives us the required street segments to reach a specified service level for each meter, is linear in the decision variables. Computational experiments showed that the linear Stage 1 IP can be optimally solved within a few seconds for large data sets. Since the Stage 2 IP is NP-complete, we developed a fast metaheuristic that generated and further improved the full route, and still maintained the specified service levels for each meter. We developed three Bayesian updating models to learn from the new incoming data in an efficient way and avoid the drawbacks faced in regression. We cross-checked our choice for the priors in the Bayesian models by comparing the parameter estimates of the logit model and the probit model with their regression counterparts. We showed

that the hierarchical probit Bayesian updating model produces more accurate probability estimates for each meter compared to the other two models. We conducted simulation experiments to compare the route qualities for the three Bayesian models and the benchmark routes used by utility companies. In our simulations, we used an actual street network and meter locations, which is different from the artificial networks used in the literature. Our simulation results showed that the routes generated by the hierarchical probit model with likelihood values of 0.75 are operationally useful because almost all meters are read with only a few miles of extra travel compared to the initial route and the total time in the two phases to read all the meters is 6.5 hours which fits into a typical 8 hour workday schedule for the drivers.

Typically, the drivers of service vehicles are more comfortable traveling through the same neighborhoods on their routes. For a utility company, this leads to meter reading routes that look similar in every period. Because of this, we do not have any information about the actual meter reading potential from most of the street segments in the network. The iterative framework proposed and discussed in this paper to generate robust routes will be more effective when we have actual meter reading data from a larger set of street segments. In future work, using Bayesian decision theory and route optimization, we would like to be able to identify new street segments to traverse at each time period so that the information gain is maximized without having to travel many extra miles.

Appendix. Remove and Repair Procedure of the Route Trimmer

In Table 10, we give the algorithm for the remove and repair procedure of the route trimmer. Lines 1-4 are the initialization steps. *HaveToBeRequired* is the subset of the required street segments that should be a part of the route. *CandidateList* is the subset of the required street segments that could potentially be removed from the route or replaced with other street segments. *RequiredSegments* is the set of required street segments that will be used by the route generator to generate the full route. *BestRoute* is the route that is feasible and shortest in length. Line 5 indicates that the algorithm will loop until the *CandidateList* is empty. Lines 6-13 represent the remove procedure. Street segment s_{mc} in the *CandidateList* with the highest marginal cost in the *BestRoute* is removed. *CandidateList* and *RequiredSegments* are updated and route $R_{s_{mc}}$ is generated using the route generator. If $R_{s_{mc}}$ is feasible, *CountInfeasible*,

Table 10: Algorithm for the remove and repair procedure of the route trimmer.

1:	$HaveToBeRequired \leftarrow M_R;$
2:	$CandidateList \leftarrow E_R \cup A_R \setminus M_R;$
3:	$RequiredSegments \leftarrow E_R \cup A_R;$
4:	$BestRoute \leftarrow$ Full route obtained using route generator;
5:	while $CandidateList \neq \emptyset$ do
6:	Find $s_{mc} \in CandidateList$ which has the highest marginal cost in $BestRoute$;
7:	$CandidateList \leftarrow CandidateList \setminus s_{mc};$
8:	$RequiredSegments \leftarrow HaveToBeRequired \cup CandidateList;$
9:	Generate full route $R_{s_{mc}}$ using route generator;
10:	if $R_{s_{mc}}$ meets the specified likelihood L_i of reading each meter i then
11:	$CountInfeasible \leftarrow 0;$
12:	if Length of $R_{s_{mc}} <$ Length of $BestRoute$ then
13:	$BestRoute \leftarrow R_{s_{mc}};$
14:	else
15:	$Success \leftarrow \text{FALSE};$
16:	$PossibleSegments \leftarrow \emptyset;$
17:	for all $s_n \notin RequiredSegments \cup s_{mc}$ do
18:	if constraint (2) is feasible for all the meters with $x_j = 1$ and $j \in$
19:	$RequiredSegments \cup s_n$ then
20:	$PossibleSegments \leftarrow PossibleSegments \cup s_n;$
21:	for all $s_p \in PossibleSegments$ do
22:	$RequiredSegments \leftarrow RequiredSegments \cup s_p;$
23:	Generate full route $R_{s_{mc}}^{s_p}$ using route generator;
24:	$RequiredSegments \leftarrow RequiredSegments \setminus s_p;$
25:	if Length of $R_{s_{mc}}^{s_p} <$ Length of $BestRoute$ then
26:	$BestRoute \leftarrow R_{s_{mc}}^{s_p};$
27:	$Success \leftarrow \text{TRUE};$
28:	if $Success = \text{FALSE}$ then
29:	$HaveToBeRequired \leftarrow HaveToBeRequired \cup s_{mc};$
30:	$CountInfeasible \leftarrow CountInfeasible + 1;$
31:	if $CountInfeasible > 10$ then
32:	break;

an infeasibility counter for a route, is set to zero, and if the length of $R_{s_{mc}}$ is smaller than the length of the $BestRoute$, $R_{s_{mc}}$ becomes the new $BestRoute$. Lines 14-30 represent the repair procedure. If $R_{s_{mc}}$ is infeasible, the repair procedure tries to add another street segment to remove the infeasibility. $Success$, a binary variable representing the success of the repair procedure, is initialized to FALSE. $PossibleSegments$ is the subset of street segments outside of $RequiredSegments$ and s_{mc} , each of which, if added to $RequiredSegments$, has the potential of generating a shorter route compared to the current $BestRoute$. If any street segment s_n , outside of the $RequiredSegments$ and s_{mc} , satisfies constraint (2) of the Stage 1 IP along with the $RequiredSegments$, then s_n is added to the $PossibleSegments$. For each street segment s_p

in *PossibleSegments*, a route $R_{s_{mc}}^{s_p}$ is generated using the route generator with the required street segments being the *RequiredSegments* and s_p . The routes $R_{s_{mc}}^{s_p}$, for each s_p , and the *BestRoute* are compared and the shortest route is set as the new *BestRoute*. If the repair procedure improves the *BestRoute*, *Success* is updated to TRUE. If *Success* is FALSE after the repair procedure, i.e., both the remove and the repair procedures involving street segment s_{mc} is unsuccessful, then s_{mc} is added to the *HaveToBeRequired* and the *CountInfeasible* is increased by one. The loop repeats by choosing a new s_{mc} from the current *CandidateList*, given that it is non-empty, based on the current *BestRoute* unless *CountInfeasible* is greater than the stopping criterion, which is set to 10.

References

- Albert JH, Chib S (1993) Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association* 88(422):669–679.
- Ávila T, Corberán Á, Plana I, Sanchis JM (2016) A new branch-and-cut algorithm for the generalized directed rural postman problem. *Transportation Science* 50(2):750–761.
- Behdani B, Smith JC (2014) An integer-programming-based approach to the close-enough traveling salesman problem. *INFORMS Journal on Computing* 26(3):415–432.
- Bodin L, Levy L (1991) The arc partitioning problem. *European Journal of Operational Research* 53:393–401.
- Colorado Department of Regulatory Agencies (2018) Public Utilities Commission. Accessed May 4, 2018, <https://www.pueblo.us/DocumentCenter/View/6596/Your-Rights-as-an-Electric-or-Natural-Gas-Utility-Customer>.
- Corberán Á, Plana I, Sanchis JM (2014) The rural postman problem on directed, mixed, and windy graphs. *Arc Routing: Problems, Methods, and Applications* (SIAM) 101–127.
- Coutinho WP, Subramanian A, do Nascimento RQ, Pessoa AA (2016) A branch-and-bound algorithm for the close-enough traveling salesman problem. *INFORMS Journal on Computing* 28(4):752–765.
- Defryn C, Sörensen K (2017) A fast two-level variable neighborhood search for the clustered vehicle routing problem. *Computers & Operations Research* 83:78–94.
- Dijkstra EW (1959) A note on two problems in connexion with graphs. *Numerische Mathematik* 1(1):269–271.
- Domencich T, McFadden DL (1975) *Urban Travel Demand: A Behavioral Analysis* (Elsevier).

- Dong J, Yang N, Chen M (2007) Heuristic approaches for a TSP variant: The automatic meter reading shortest tour problem. *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies* (Springer Verlag) 145–163.
- Dumitrescu A, Mitchell J (2003) Approximation algorithms for TSP with neighborhoods in the plane. *Journal of Algorithms* 48:135–159.
- Eglese R, Golden B, Wasil E (2014) Route optimization for meter reading and salt spreading. *Arc Routing: Problems, Methods, and Applications* (SIAM) 300–320.
- Frederickson GN, Hecht MS, Kim CE (1978) Approximation algorithms for some routing problems. *SIAM Journal on Computing* 7(2):178–193.
- Groër C, Golden B, Wasil E (2009) The balanced billing cycle vehicle routing problem. *Networks* 54(4):243–254.
- Gulczynski D, Heath J, Price C (2006) The close enough traveling salesman problem: A discussion of several heuristics. *Perspectives in Operations Research: Papers in Honor of Saul Gass' 80th Birthday* (Springer Verlag) 271–283.
- Hà MH (2012) Modélisation et résolution de problèmes généralisés de tournées de véhicules, Ph.D. dissertation, Automatique, Ecole des Mines de Nantes, France.
- Hà MH, Bostel N, Langevin A, Rousseau L-M (2014) Solving the close-enough arc routing problem. *Networks* 63(1):107–118.
- Hansen P, Mladenović N, Todosijević R, Hanafi S (2017) Variable neighborhood search: Basics and variants. *EURO Journal on Computational Optimization* 5(3):423–454.
- Hastings WK (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57:97–109.
- Illinois Administrative Code (2018) Title 83: Public Utilities, Section 280.90: Estimated Bills. Accessed May 4, 2018, <ftp://www.ilga.gov/JCAR/AdminCode/083/083002800F00900R.html>.
- Irving, Texas - Code of Ordinances (2018) Chapter 31: Public Utilities, Article II: Natural Gas, Section 31-11: Estimated bills prohibited. Accessed May 4, 2018, https://library.municode.com/tx/Irving/codes/code_of_ordinances?nodeId=PTIITHCO_CH31PUUT_ARTIINAGA_S31-11ESBIPRCOEXTESE.
- Kwon O, Golden B, Wasil E (1995) Estimating the length of the optimal TSP tour: An empirical study using regression and neural networks. *Computers & Operations Research* 22(10):1039–1046.
- Levy L, Snizek J, Cox B (2002) Utility meter route management and optimization using GIS, Tech. report, Electric and Gas Utilities User Group, Coeur d’Alene, Idaho.

- Louviere JJ, Hensher DA, Swait JD (2000) *Stated choice methods: Analysis and applications* (Cambridge University Press).
- Lum O, Golden B, Wasil E (2018) An open-source desktop application for generating arc-routing benchmark instances. *INFORMS Journal on Computing* 30(2):361-370.
- Marin J-M, Robert CP (2014) *Bayesian Essentials with R* (Springer).
- Mennell WK (2009) Heuristics for solving three routing problems: Close-enough traveling salesman problem, close-enough vehicle routing problem, sequence-dependent team orienteering problem, Ph.D. dissertation, Decision, Operations & Information Technologies, University of Maryland, College Park, USA.
- Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of state calculations by fast computing machines. *Journal of Chemical Physics* 21:1087-1091.
- Michigan Department of Labor and Economic Growth (2018) Public Service Commission, Consumer Standards and Billing Practices, Part 4. Accessed May 4, 2018, <https://www.consumersenergy.com/~media/CE/Documents/mpsc-billing-rules.ashx?la=en>.
- Press SJ (2003) *Subjective and Objective Bayesian Statistics: Principles, Models, and Applications* (John Wiley & Sons).
- Renaud A, Absi N, Feillet D (2017) The stochastic close-enough arc routing problem. *Networks* 69(2):205-221.
- Rossi PE, Allenby GM, McCulloch R (2005) *Bayesian Statistics and Marketing* (John Wiley & Sons).
- Shuttleworth R, Golden B, Smith S, Wasil E (2008) Advances in meter reading: Heuristic solution of the close enough traveling salesman problem over a street network. *The Vehicle Routing Problem: Latest Advances and New Challenges* (Springer Verlag) 487-501.
- Steiniger S, Hunter AJS (2013) The 2012 free and open source GIS software map - A guide to facilitate research, development, and adoption. *Computers, Environment and Urban Systems* 39:136-150.
- Stern D, Dror M (1979) Routing electric meter readers. *Computers & Operations Research* 6:209-223.
- Wassan N, Wassan N, Nagy G, Salhi S (2017) The multiple trip vehicle routing problem with backhauls: Formulation and a two-level variable neighbourhood search. *Computers & Operations Research* 78:454-467.

Online Appendix. Analysis and Insights from a Smaller Data Set

A smaller data set was gathered during the second half of 2015 by ITRON and provided by RouteSmart Technologies. This data set gives meter locations and reading data in Liberty Lake, Washington. A summary of the data set is given in Table A.1.

A Detailed Data Analysis

In Figure A.1, we use box plots to show the relationship between the minimum time interval between reads and the number of reads. Consider the value of three on the x -axis, i.e., the number of reads is three. We are considering meters in the service location layer that are read exactly three times from the route traversed by the meter reading vehicle (there are 29 such meters). For these meters, we have a time interval between their first read and second read, and a time interval between their second read and third read. We take the minimum of these

Table A.1: Summary of the data set.

Total number of meters in the service location layer	474
Number of meters in the service location layer that are read	209
Total number of read events	28,745
Number of read events from meters in the service location layer	827
Number of street segments traversed in the route	7
Time gap between consecutive signal transmission (sec)	13
Maximum read distance among all meters in the service location layer (feet)	3,510

Figure A.1: Minimum time interval between reads.

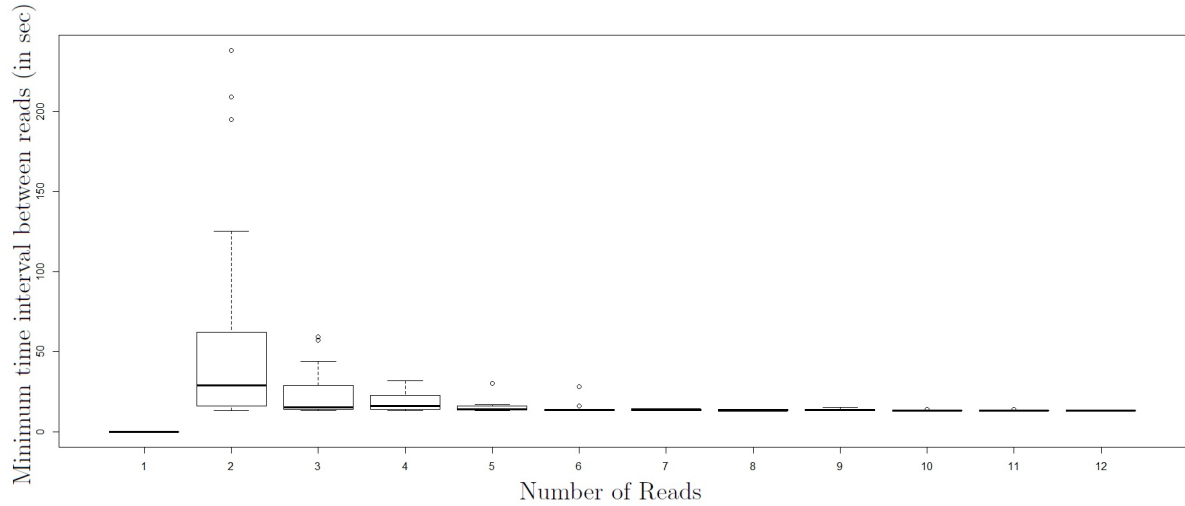
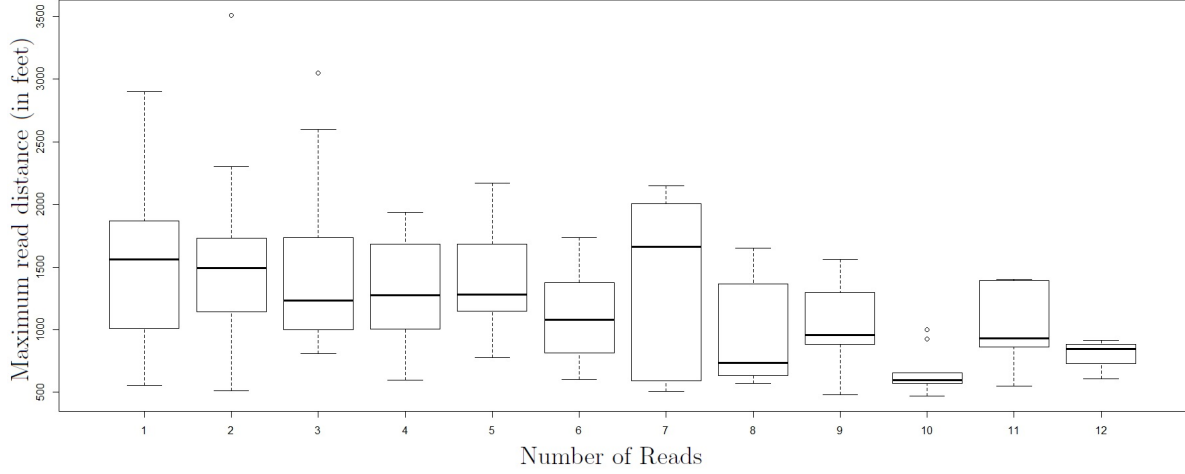


Figure A.2: Maximum read distance.



two time intervals for each of the 29 meters. The 29 minimum time interval values, which range from 13 seconds to 59 seconds, are shown using a box plot at $x = 3$. We observe that most of the large values of the minimum time interval occur with a fewer number of reads. When the number of reads is less for a meter, the vehicle is probably farther from the meter most of the time on the route. The vehicle probably came close to the meter for small portions of the route. When the number of reads is larger for a meter, the vehicle is probably closer to the meter for a longer duration. The vehicle should have read that meter every second. However, this is not the case. Instead, for this data set, the minimum time intervals attain a constant value of 13 seconds. This value is the time gap between consecutive signals sent by the RFID transmitters.

In Figure A.2, we use box plots to show the relationship between the maximum read distance and the number of reads. Again, consider the value of three on the x -axis. For the 29 meters, we have the read distances for each of their three reads. We take the maximum of these three read distances for each of the meters. These 29 maximum read distance values, which range from 808 feet to 3,052 feet, are shown using a box plot at $x = 3$. From our observations, it seems that the chances of having larger values of the maximum read distance increases for meters with a smaller number of reads. This also confirms our observations from Figure A.1. When the number of reads is less for a meter, the vehicle is farther from the meter most of the time on a route; when the number of reads is larger for a meter, the vehicle is closer to the meter for a longer duration.

Table A.2: Results based on the analysis from Step 4.

Radius (feet)	Cumulative Success	Non-Cumulative Success
100	14/14 = 1.00	14/14 = 1.00
200	35/35 = 1.00	21/21 = 1.00
300	53/54 = 0.98	18/19 = 0.95
400	64/67 = 0.96	11/13 = 0.85
500	74/78 = 0.95	10/11 = 0.91
600	85/94 = 0.90	11/16 = 0.69
700	97/108 = 0.90	12/14 = 0.86
800	117/131 = 0.89	20/23 = 0.87
900	129/147 = 0.88	12/16 = 0.75
1000	149/171 = 0.87	20/24 = 0.83

We perform additional analysis on the data set with respect to a meter being read or not being read. In the first step, the route traversed by a meter reading vehicle is discretized using the distinct geographic coordinates of the vehicle’s location during the read events. In the second step, the shortest distance of the meters from the route traversed by a vehicle is calculated using the distinct locations of the vehicle. In the third step, around each of the distinct points in the discretized route, a circular disc (radius of 100 feet to 1000 feet with steps of 100) is considered. In the fourth step, for each radius, we count the number of meters within at least one of the circular discs and the number of meters that are read (regardless from where the meters are read).

In Table A.2, we provide the results based on the analysis from the four steps described above. The fraction of meters read are calculated both cumulatively and non-cumulatively for each of the 10 different radii, ranging from 100 feet to 1000 feet. The entries in the two columns have the form a/b , where b denotes the number of meters within that radius for the cumulative case and the number of meters between that radius and the preceding lower radius considered for the non-cumulative case, and a denotes the number of meters read out of those b meters. The fractions in the cumulative case show a gradual decrease in success with an increase in the distance of meters from the route. We do not observe any specific trend for the non-cumulative case. In the non-cumulative case, we note that the smallest value of the fraction occurs for meters that are at a distance of 500 feet to 600 feet from the route. This observation indicates that the shortest distance of meters from routes is not the only key factor for reading a meter successfully. Otherwise, the non-cumulative case would have followed the same trend as the cumulative case.

B Regression and Bayesian Updating

We constructed six logistic regression models and six probit regression models. For the logistic and probit regressions, Model 1 uses three independent variables: $\text{Shortest_Distance}_{ij}$, No_of_Pulses_j , and No_of_Customers_i . Model 2 adds indicator variables for the traversed street segments to Model 1. Model 3 adds indicator variables for the meters to Model 1. Model 4 adds indicator variables for the traversed street segments to Model 3. Model 5 uses $\text{Shortest_Distance}_{ij}$, No_of_Pulses_j , and indicator variables for the meters. Model 6 adds indicator variables for the traversed street segments to Model 5. The data used for estimating the parameters is $N = 474 \text{ (number of meters)} \times 7 \text{ (number of traversed street segments)} = 3,318$.

In Tables B.1 and B.2, we present the logistic regression results and the probit regression results, respectively. We give the means of the coefficients of the independent variables and their standard deviations in parenthesis. The results in Tables B.1 and B.2 for the six models are similar for both logistic regressions and probit regressions. For each of the six models in both regressions, the coefficient of Shortest_Distance is always significant and negative; the coefficient of No_of_Pulses whenever significant is positive; the coefficient of No_of_Customers whenever significant is negative. In Figures B.1 and B.2, we give the histograms for the fitted values of the dependent variable from the logistic regression models and the probit regression models, respectively. The histograms for each of the six logistic regression models are similar to the histograms for each of the six respective probit regression models.

In Tables B.1 and B.2 for both regressions, all three independent variables are significant at the 1% level in Model 1. In Model 2, for both regressions, No_of_Pulses is not significant, and the other two independent variables are significant at the 1% level in the presence of street dummies. In Model 3, for both regressions, No_of_Customers is not significant, and the other two independent variables are significant at the 1% level in the presence of customer dummies. In Model 4, for both regressions, No_of_Customers is not significant, Shortest_Distance is significant at the 1% level, and No_of_Pulses is significant at the 5% level and the 10% level for logistic regression and probit regression, respectively, in the presence of both street dummies and customer dummies. In two of the first four models, No_of_Customers is not significant, so we construct two models with customer dummies that leave out No_of_Customers . In Model 5, for both regressions, the two independent variables are significant at the 1% level in the presence of

Table B.1: Logistic regression results.

	Read_OR_Not					
	(1)	(2)	(3)	(4)	(5)	(6)
Shortest_Distance	-0.003*** (0.0002)	-0.003*** (0.0002)	-0.007*** (0.0005)	-0.007*** (0.0010)	-0.007*** (0.0005)	-0.007*** (0.0010)
No_of_Pulses	0.060*** (0.010)	-0.004 (0.053)	0.091*** (0.016)	0.165** (0.080)	0.091*** (0.016)	0.165** (0.080)
No_of_Customers	-0.017*** (0.005)	-0.021*** (0.005)	0.060 (0.204)	0.057 (0.202)		
Constant	2.852*** (0.258)	3.680*** (0.382)	8.137 (6.035)	7.983 (5.951)	9.577*** (1.669)	9.357*** (1.641)
Street Dummies	No	Yes	No	Yes	No	Yes
Customer Dummies	No	No	Yes	Yes	Yes	Yes
McFadden's Adjusted R ²	0.514	0.525	0.377	0.381	0.377	0.381

*p<0.1; **p<0.05; ***p<0.01

customer dummies. In Model 6, for both regressions, Shortest_Distance is significant at the 1% level, and No_of_Pulses is significant at the 5% level and the 10% level for logistic regression and probit regression, respectively, in the presence of both street dummies and customer dummies. Based on the results in Tables B.1 and B.2, Model 1 and Model 2 perform the best for both logistic regression and probit regression, with Model 2 performing slightly better than Model 1. The simplicity of Model 1 without any dummies, however, makes it preferable to Model 2.

Table B.2: Probit regression results.

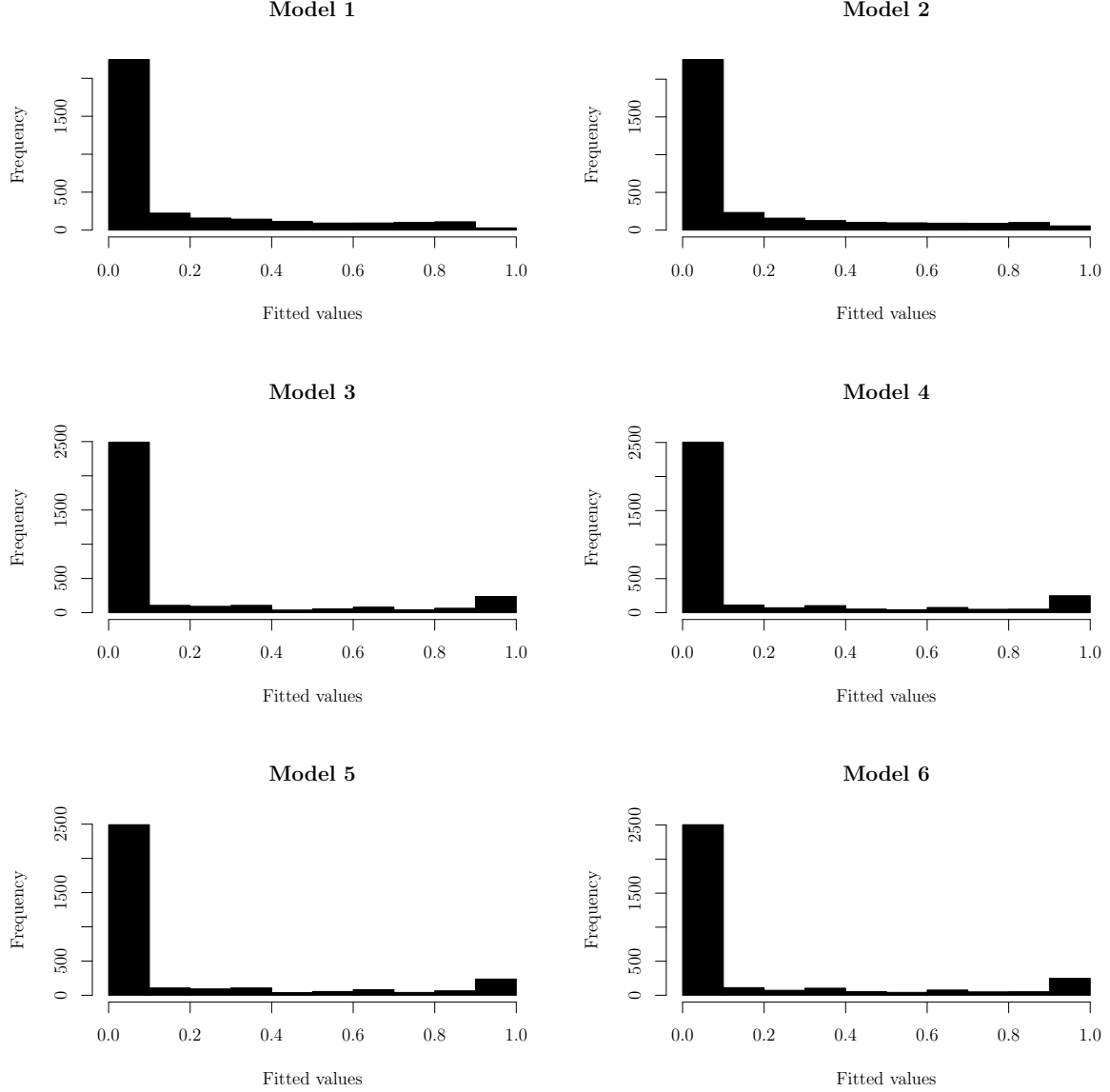
	Read_OR_Not					
	(1)	(2)	(3)	(4)	(5)	(6)
Shortest_Distance	-0.002*** (0.0001)	-0.002*** (0.0001)	-0.004*** (0.0002)	-0.004*** (0.0003)	-0.004*** (0.0002)	-0.004*** (0.0003)
No.of_Pulses	0.032*** (0.005)	-0.009 (0.030)	0.052*** (0.009)	0.084* (0.044)	0.052*** (0.009)	0.084* (0.044)
No.of_Customers	-0.013*** (0.003)	-0.015*** (0.003)	0.032 (0.111)	0.032 (0.110)		
Constant	1.721*** (0.147)	2.180*** (0.213)	4.479 (3.411)	4.378 (3.377)	5.254*** (0.986)	5.138*** (0.977)
Street Dummies	No	Yes	No	Yes	No	Yes
Customer Dummies	No	No	Yes	Yes	Yes	Yes
McFadden's Adjusted R ²	0.512	0.523	0.377	0.381	0.377	0.381

*p<0.1; **p<0.05; ***p<0.01

Model 1 has McFadden's Adjusted R² values of 0.514 and 0.512 for logistic regression and probit regression, respectively. These values indicate very good model fits. We also tried models with higher powers of independent variables and with interactions between independent variables but none of them performed better than Model 1 for both logistic regression and probit regression.

In Tables B.3 and B.4, we give the logit model results and the probit model results, respectively. The mean and standard deviation of the β_i 's from the MH random walk algorithm and

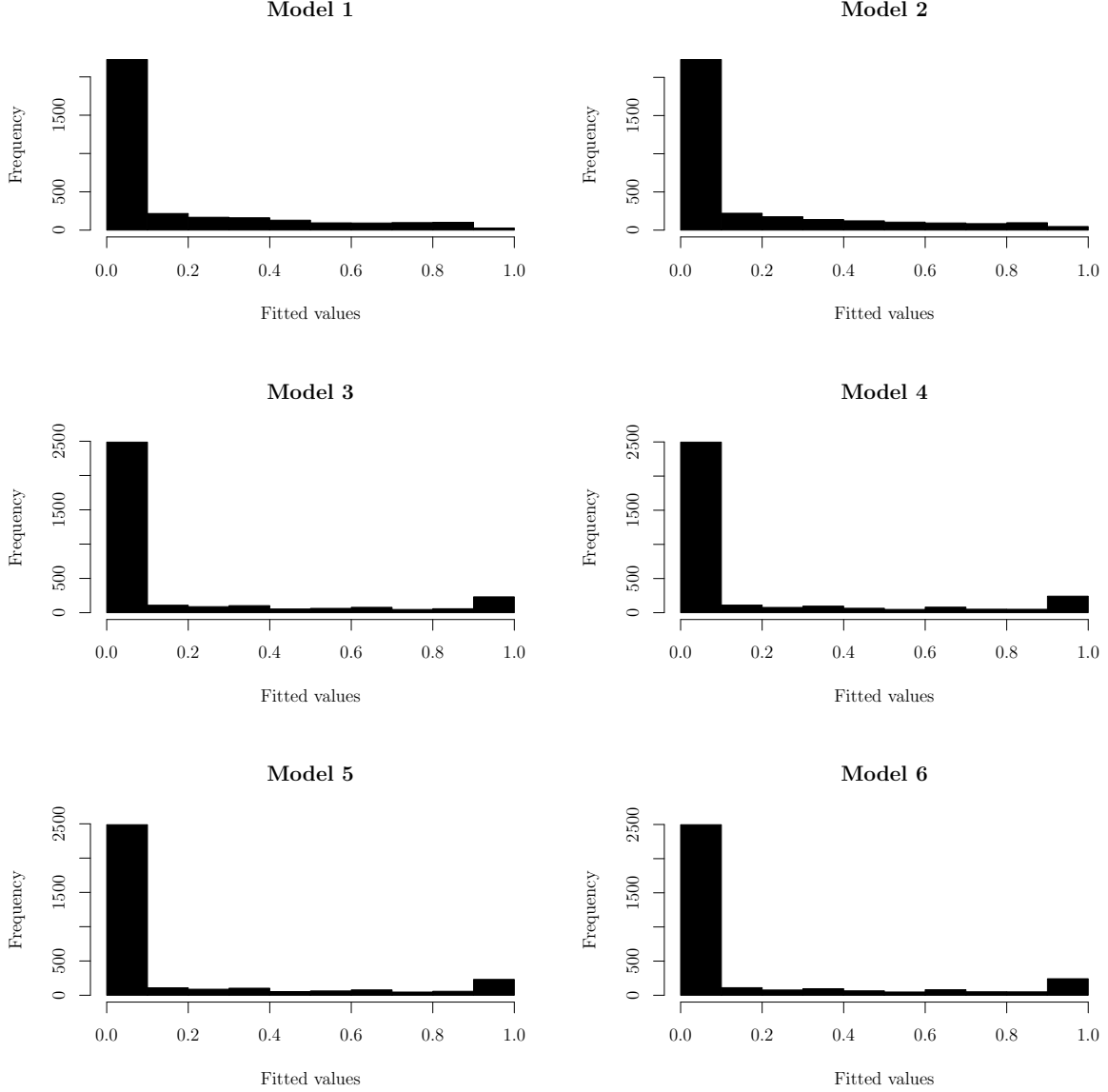
Figure B.1: Histograms of fitted values from the logistic regressions.



logistic regression, and the Gibbs sampling algorithm and probit regression are presented. Since, for each i , β_i values match for both the logistic regression and the MH random walk algorithm, and both the probit regression and the Gibbs sampling algorithm, our choice of the prior in both algorithms serves the purpose of a vague prior. This indicates that we can perform Bayesian updating for the logit model and the probit model after receiving new data points instead of using logistic regression or probit regression, respectively.

In Table B.5, we give the hierarchical probit model results for Θ . The mean and standard

Figure B.2: Histograms of fitted values from the probit regressions.



deviation of the $\theta_{i,j}$'s from the Gibbs sampling algorithm are presented. In Figure B.3, we show the histograms of the means of the β_i 's from the Gibbs sampling algorithm for the hierarchical probit model. The three histograms belong to the coefficients of the Intercept ($\hat{\beta}_{i,1}$), the Shortest_Distance ($\hat{\beta}_{i,2}$), and the No_of.Pulses ($\hat{\beta}_{i,3}$), respectively, for the lower level probit model in the hierarchical probit model. The histograms show the variation in the coefficients for different meters that are not captured in the logit model or the probit model. The hierarchical probit model gives individualized probability predictions p_{ij} for each meter i . The size of the data sets

Table B.3: Logit model results.

Coefficient	Logistic Regression	MH Random Walk Algorithm
Intercept (β_1)	2.852 (0.258)	2.860 (0.267)
Shortest_Distance (β_2)	-0.003 (0.0002)	-0.003 (0.0001)
No_of_Pulses (β_3)	0.060 (0.010)	0.060 (0.009)
No_of_Customers (β_4)	-0.017 (0.005)	-0.018 (0.005)

Table B.4: Probit model results.

Coefficient	Probit Regression	Gibbs Sampling Algorithm
Intercept (β_1)	1.721 (0.147)	1.733 (0.166)
Shortest_Distance (β_2)	-0.002 (0.0001)	-0.002 (0.0001)
No_of_Pulses (β_3)	0.032 (0.005)	0.032 (0.005)
No_of_Customers (β_4)	-0.013 (0.003)	-0.013 (0.003)

used for estimating the lower level parameters in the probit models for each meter i is $m = 7$. The size of the data set used for estimating the higher level parameters in the multivariate linear regression model is $n = 474$.

Table B.5: Hierarchical probit model results for the higher level parameter matrix.

Coefficient	Gibbs Sampling Algorithm
$\theta_{1,1}$	11.685 (4.008)
$\theta_{2,1}$	-0.010 (0.009)
$\theta_{3,1}$	-0.101 (0.090)
$\theta_{1,2}$	0.013 (0.061)
$\theta_{2,2}$	-0.0002 (0.0002)
$\theta_{3,2}$	0.005 (0.002)

Figure B.3: Histograms of the means of the lower level parameters from the Gibbs sampling algorithm for the hierarchical probit model.

